(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
18 October 2001 (18.10.2001)

PCT

(10) International Publication Number
WO 01/77841 A2

(51) International Patent Classification[7]: G06F 13/00

(21) International Application Number: PCT/US01/11505

(22) International Filing Date: 9 April 2001 (09.04.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

09/545,067	7 April 2000 (07.04.2000)	US
09/544,754	7 April 2000 (07.04.2000)	US
09/545,337	7 April 2000 (07.04.2000)	US

(71) Applicant: NETWORK APPLIANCE, INC. [US/US]; 495 East Java Drive, Sunnyvale, CA 94089 (US).

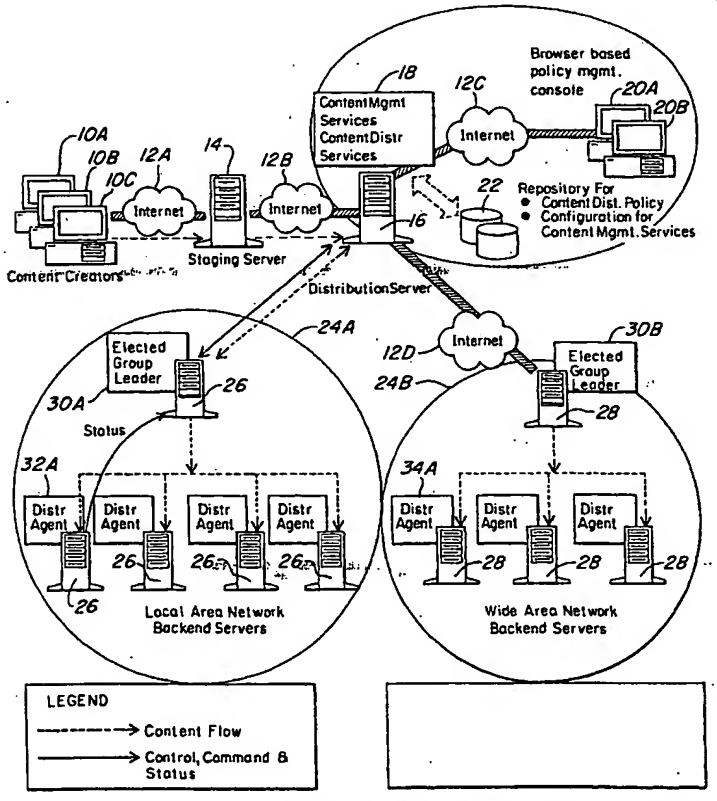
(72) Inventors: BASANI, Vijay, R.; 26 Kessler Farm Drive, #418, Nashua, NH 03062 (US). MANGIAPUDI, Krishna; 5 Decatur Drive, Nashua, NH 03062 (US). MURACH, Lynne, M.; Bramble Hill Road, Methuen, MA 01844 (US). KARGE, Leroy, R.; 400 White Pond Road, Leominster, MA 01453 (US). REVSIN, Vitaly, S.; 4 Enfield Drive, Andover, MA 01810 (US). BESTAVROS, Azer; 46 Rice Road, Wayland, MA 01778 (US). CROVELLA, Mark, E.; 14 Collier Road, Scituate, MA 02066 (US). LAROSA, Domenic, J.; 16 Meditation Lane, Atkinson, NH 03062 (US).

(74) Agents: NELSON, Barry, C. et al.; Brown Rudnick Freed & Gesmer, One Financial Center, Boston, MA 021111 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

[Continued on next page]

(54) Title: METHOD AND APPARATUS FOR RELIABLE AND SCALABLE DISTRIBUTION OF DATA FILES IN DISTRIBUTED NETWORKS



(57) Abstract: The present invention provides a system and apparatus for efficient and reliable, control and distribution of data files or portions of files, applications, or other data objects in large-scale distributed networks. A unique content-management front-end provides efficient controls for triggering distribution of digitized data content to selected groups of a large number of remote computer servers. Transport-layer protocols interact with distribution controllers to automatically determine an optimized tree-like distribution sequence to group leaders selected by network devices at each remote site. Reliable store-and-forward transfer to clusters is accomplished using a unicast protocol in the ordered tree sequence. Once command messages and content arrive at all participating group leaders, local hybrid multicast protocols efficiently and reliably distribute them to the back-end nodes for interpretation and execution. Positive acknowledgement is then sent back to the content manager from each group leader, and the updated content in each remote device autonomously goes "live" when the content change is locally completed.

(84) **Designated States** *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:

— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

METHOD AND APPARATUS FOR RELIABLE AND SCALABLE DISTRIBUTION OF DATA FILES IN DISTRIBUTED NETWORKS

Field Of The Invention

This invention is directed towards data communication, and more particularly towards reliable and efficient distribution of data to large numbers of network locations.

Background Of The Invention

Digital content creators are users who utilize workstations or other computers to create or digitize information in preparation for publication as "content." When such content is to be shared with or published to a number of other computer users using a wide area network (WAN), such as the World Wide Web ("the Web"), reliability, latency, security, and efficiency become major issues. Reliability refers to the ability to ensure that the data was received without debilitating errors. Latency, the measure of how much time it takes to deliver data, suffers when finite resources become overloaded, whether in the respective processors, intermediate storage or a communications link. Inefficiency may arise because multiple copies of data have to be retransmitted between the same source(s) and destination(s) due to lost or garbled messages. As the number of recipient sites grows, issues of latency and efficiency complicate the architecture.

Inefficient communication protocols for reliable data exchange amplify problems in real-time systems where latency directly determines user satisfaction.

Historically, manual or customized operations were the only solutions available for distributing new or modified content, as networks expanded and data-distribution needs changed.

However, such solutions have the disadvantage of not being flexible enough to handle real-time load balancing. Temporary outages of system components can also cause havoc in a statically defined distribution method. Similarly, manual or customized actions become increasingly labor-intensive as data files proliferate and the number of servers increases exponentially, as seen in the recent growth of the Internet. In particular, the

operation of the "Web" requires massive data management and distribution. Many users expect instantaneous access, worldwide, to the fastest source of the best data available at any given moment. This puts a heavy burden on service providers for better information control and infrastructure management.

One well known solution to reduce access latency by large numbers of users is to distribute content to file servers at numerous remote sites, and then direct user access requests to those servers. Multiple copies of content must then be tracked and synchronized in order to provide uniformity and consistency of data among all users. Many network content publishers obtain network file server services from a variety of geographically dispersed service providers. Manual coordination with each service provider for content distribution increases complexity and creates more room for error and delay.

To manage the problem of rapid content distribution from a master copy, several companies have experimented with or proposed semi-automated systems for streamlining the distribution process. These solutions are typically targeted at one of three critical points: "content management;" reliable and efficient distribution across WANs; or the local replication and synchronization across multiple servers with1. (Amended) A system for distributing information to a set of destination nodes connected via a communication network comprising:

a reporting process in each destination node for generating and transmitting a report to a distribution manager, said report containing an identification of said destination node and corresponding destination node parameters, whereby said destination node offers to become a participant in a distribution job;

said distribution manager connected to said network and configured to receive said reports from said destination nodes and to create a prioritized list of destination nodes selected as participants in a distribution job according to said destination node parameters, and having a management process for sending information to each participant;

said management process adapted to send each participant instructions to obtain a copy of said information either from said distribution manager or from another identified participant, until each participant has received a copy of said information;

each participant having a store-and-forward process configured to receive instructions from a prior participant or from said distribution manager and to request a

copy of said information from said prior participant or from said distribution manager, and to thereafter request further distribution instructions from said distribution manager until instructed that no other participants require said information;

whereby each participant obtains a copy of the information and the distribution manager obtains confirmation that each destination node has obtained said information.

One example of a content management system is the Content Delivery Suite (CDS) product distributed by Inktomi Corporation of Foster City, California, as described at www/inktomi.com/products/network/traffic/tech/cdswhitepaper. According to the available documentation, CDS management components determine when data content changes within file systems on a "staging server," and then send updated files to "CDS Agents" on distributed web-servers. Once the updated files are received at the web servers, the CDS triggers all web servers to take the updated files "live" simultaneously. This particular solution suffers from numerous disadvantages. Sending entire files for an update is relatively inefficient, when only a small amount of data may have actually changed out of millions of bytes in the file. File transmission to each remote server originates from a single, central point, and all remote servers must wait for the others accessing the same central source to receive and acknowledge the correct data before the new content goes "live." The referenced implementation lacks the ability to intelligently schedule distribution or replication of pertinent content to different parts of the network according to the user's needs.

Another example of a system for managing content distribution is the global/SITE product of F5 Networks, Inc., of Seattle, WA., as described at http://www.f5.com/globalsite/index.html. The available documentation indicates that global/SITE is an additional computer appliance that is added to a LAN and a central site. The specialized hardware and software at the central site automatically replicates and transfers only those files that have changed (i.e., new, updated, or deleted). Changes to updated files include only the changed portions, thus reducing the wasted transmission load. However, disadvantageously, the addition of separate hardware and software at each site inherently reduces reliability, since there are more components subject to maintenance and potential failure. In fact, the global/SITE system becomes a single point of failure which could cripple an entire site if the unit is rendered inoperable, whether accidentally

or maliciously. Installation, configuration and maintenance of these additional units will also require on-site support and customized spare parts.

One approach to schedule management is proposed in U.S. Pat. No. 5,920,701 ("the '701 patent"), issued July 6, 1999. The '701 patent teaches a system in which data transfer requests and schedules from a content source are prioritized by a network resource scheduler. Based upon the available bandwidth and the content priority, a transmission time and data rate is given to the content source to initiate transmission. The scheduler system requires input that includes information about the network bandwidth, or at least the available bandwidth in the necessary content path. This has the disadvantage of requiring additional complexity for determination of network bandwidth at any given moment. It also requires a method for predicting bandwidth that will be available at some transmission time in the future. Furthermore, a content distributor is required to provide a "requested delivery time deadline," which complicates content management by requiring each content distribution requester to negotiate reasonable transmission times for each piece of content. This approach is focused entirely on bandwidth allocation, and fails to address issues of network dynamics, such as regroupings of the target servers for load-balancing. Whatever efficiency may have been derived from the '701 is substantially completely lost when the entire content must be retransmitted to an additional server, making a huge waste of bandwidth for every node in the multicast path which already received the file.

Each of these alleged management and distribution solutions relies upon file replication and transmission techniques that remain closely tied to one-on-one file transfers to each individual server. The problem grows geometrically as the number of servers increases and multiple copies of selected files are required at each remote web site.

The ubiquitous Internet Protocol (IP) breaks messages into packets and transmits each one to a router computer that forwards each packet toward the destination address in the packet, according to the router's present knowledge of the network. Of course, if two communicating stations are directly connected to the same network (e.g., a LAN or a packet-switching network), no router is necessary and the two stations can communicate directly using IP or any other protocol recognized by the stations on the network. A "web farm" or "cluster" is an example of a LAN on which multiple servers are located. In a

cluster, there is typically a front-end connected to the Internet, and a set of back-end servers that host content files.

LANs, by their nature, are limited in their ability to span long distances without resorting to protocol bridges or tunnels that work across a long-distance, point-to-point link. Since most LAN protocols were not designed primarily for Wide Area Networking, they have features that can reduce reliability and efficiency of the LAN when spanning a WAN. For example, a station on a LAN can send a multicast IP packet simultaneously to all or selected other stations on its LAN segment very efficiently. But when the LAN is connected to an IP router through the Internet to another router and other LAN segments, the multicast becomes difficult to manage, and reliability suffers. In particular, most Internet routers only handle point-to-point, store-and-forward packet requests and not multicast packet addresses. This puts the burden on the sender to laboriously transmit a separate copy to each intended remote recipient, and to obtain a positive acknowledgement of proper receipt.

One proposed solution, described in U.S. Pat. No. 5,727,002, issued March 10, 1998, and in U.S. Pat. No. 5,553,083, issued September 3, 1996, relies upon the limited multicast capabilities of IP to reach large numbers of end-points with simultaneous transmissions. Messages are broken into blocks, and blocks into frames. Each frame is multicast and recipients post rejections for frames not received, which are then retransmitted to the multicast group until no further rejections are heard. A disadvantage of the disclosed method is that it relies upon either a network broadcast of data at the application layer, or a multicast IP implementation based upon the standardized RFC 1112 Internet specification. Broadcast is an extremely inefficient protocol in all but very limited circumstances, since it requires that each and every recipient process an incoming message before the recipient can determine whether or not the data is needed. Even multicast IP has the disadvantage of being based upon the unwarranted assumption that the Internet routers will support the standard multicast feature, which is actually very rare.

Under limited condition, i.e., where the Internet routers actually support the IP multicast feature, a packet can be sent simultaneously to many receivers. Building upon IP multicast, Starburst Software, Inc., of Concord, MA (the assignee of the '002 and `083 patents mentioned above), has created the Starburst OmniCast product, described at http://www.starburstsoftware.com/products/omnicast3.pdf and in a Starburst Technology

Brief. As described, the OmniCast product relies upon the router to replicate and forward the data streams to multiple destinations simultaneously. This has the disadvantage of not being applicable to most of the present Internet, or in any private network that does not implement multicast according to the standard. Alternatively, using a so-called "FanOut" feature, the OmniCast application itself replicates the packets and forwards them to multiple FanOut sites which then use local multicast features for further distribution. Each FanOut server is configured to accept certain multicast addresses. The FanOut closest to the source replicates the packets and sends them to a configured list of addresses using a unicast protocol, and encapsulates the multicast address for further use by downstream FanOuts. This solution has the disadvantage of requiring configuration and maintenance of static lists of servers in each FanOut unit. It also does not provide any flexibility for changing which back-end servers correspond to each multicast address. The central FanOut unit is also burdened with sequential transmission of the first message to every remote FanOut unit, using a unicast protocol.

Another disadvantage of existing implementations is that they fail to deal with much of the dynamic nature of the Internet, in which servers are reallocated from time to time, or new servers are added for performance considerations. Current implementations rely upon manual, error-prone coordination between groups of personnel who create content and those who manage the network resources.

Some large-scale distributed networks use processor group leaders to manage distribution and communication of data. However, disadvantageously, group leaders can be lost, such as when the system providing that service is taken offline or otherwise becomes unavailable. In one approach to recovery of a group leader in a distributed computing environment, described in U.S. Pat. No. 5,699,501, issued December 16, 1997, a system of servers has a group leader recovery mechanism in which a new group leader can be selected from a list of servers, in the order in which processors join the group. The list is distributed via multicast or held in a name server, and is accessed whenever a new group leader is needed. The disadvantage of this approach is that each server has the same chance of becoming the leader, even though there may be numerous reasons to make a better selection.

Another disadvantage of existing systems is that load-balancing processes or service-level monitors, that may be operating simultaneous with content distributors,

typically have no way to directly determine whether a particular server has the most recent version of content. Similarly, in situations where content is transparently cached in alternate servers, someone has to remember to update (i.e., purge) the cache when there are changes to the cache. Most cache implementations also have no capability for making efficient updates when changes are small in proportion to the size of the file containing the changes.

Summary Of The Invention

The present invention provides a method and apparatus for efficient and reliable control and distribution of data files or portions of files, applications, or other data objects in large-scale distributed networks. A unique content-management front-end provides efficient controls for triggering distribution of digitized data content to selected groups of a large number of remote computer servers. Transport-layer protocols interact with distribution controllers to automatically determine an optimized tree-like distribution sequence to group leaders selected by network devices at remote sites. Reliable transfer to clusters is accomplished using a unicast protocol in the ordered tree sequence. Once packets arrive at the remote cluster, local hybrid multicast protocols efficiently and reliably distribute them to the back-end nodes for interpretation and execution. Positive acknowledgement is then sent back to the content manager from each cluster, and the updated content in each remote device autonomously goes "live" when the content change is locally completed.

According to the present invention content creators deposit digitized data content on a staging server on the network, for example via the Internet. The staging server and distributions servers can be physically separate computers or could both reside on the same computer. The staging server is interrogated by a distribution server running a content management service known as content control manager ("CCM"), according to configurable policies (such as scheduled updates, events, backups). A browser-based policy management system interacts with the distribution server to establish content management service configurations and content distribution policies. Scheduled content transactions (such as updates, synchronization, replications, backups, restorations, or rollback) are monitored by a scheduler to avoid server conflicts and to minimize network

congestion. The scheduler detects scheduled job conflicts and notifies the user to reschedule a job. When a content transaction (or "job") is initiated, a set of necessary directory and file changes are determined, according to configurable policies, along with the commensurate steps needed to carry out the job, known as "assignments."

The content control manager issues assignments to system components for creating or deleting remote server directories and files, and for distributing changed content from the staging server. Remote servers are administratively divided into "content groups." Content Groups are logical groupings of remote servers that will participate in or receive the content distribution, either within a LAN or across WANs. Assignments, which comprise assignment commands and the content data, are then forwarded to dynamically configured cluster Group Leaders ("GLs"). The Group Leader is responsible for overseeing the distribution of the assignment to the remote or BackEnd Servers ("BESs") that are in the Content Group within the GLs network segment. A component on the BES receives and processes the assignment, reporting success or failure back to the Group Leader. The Group Leaders each report the status of the assignment for all of their corresponding BESs to the CCM. The CCM reports the assignment status back to the database and optionally logs the status to a log file. The status can be viewed through the browser-based User Interface. Completed assignments are reported directly to the database, along with the completion status. Failed assignments are rescheduled (or cancelled) according to the current database policies for the corresponding content.

In further accord with the invention, an assignment message contains instructions for creating, moving/copying, removing, or modifying directories or file content on remote servers, including parameters for any required compression and encryption. The assignment itself, or any of its components, can be encrypted prior to transmission to provide for enhanced security, including privacy or integrity, or both. Assignments are dispatched according to a sorted list of group leaders, based on factors such as nearness, processor speed, reliability, or CPU Usage, and according to the content groupings. For a small number of GLs, each GL can be individually and directly addressed by the CCM. However, as the number of network segments grows, a store-and-forward approach becomes much more efficient. According to a distribution mechanism for storing and forwarding content among group leaders, the first selected group leader receives the first assignment from the content control manager (CCM). Before or while carrying out its

own assignment, the first group leader (GL) requests instructions for the next GL on the list from the CCM and forwards that assignment to the next GL. Each GL in turn handles its own assignment for its cluster, reports its status, requests the next GL's assignment from the CCM, and forwards the assignment to the next GL. When all the GLs have received the assignment, the GLs distribute the assignment to their corresponding BESs. This mechanism permits highly efficient and robust distribution of assignments and data content from the CCM to each required GL using a store-and-forward tree structure.

In further accord with a mechanism for distributing content to dynamically elected group leaders, a dynamic tree structure is maintained by the system based upon real-time nominations of GLs and their respective registration of group members within each cluster, reported to and processed by the CCM. The members of a group elect a group leader according to real-time and administration selection criteria. The elected GL then reports its registered group membership and performance parameters to the CCM. The CCM processes these reports and derives an optimally sorted list of GLs for distribution of assignments. The list of clusters for distribution of assignments is arranged in an order according to dynamic network factors such as location. There is a user interface mechanism to allow a system administrator to override (or configure) this election and arrangement behavior and to artificially define a static behavior.

In further accord with the invention, once a GL has received an assignment destined for its own members, and no further GLs require distribution of the assignment, each GL uses a reliable Multicast Content Transport Protocol (MCTP) to distribute the assignment to each of the BESs in the addressed group. Once the BES receives the assignment, a Content Interpreter (CI) parses the assignment and carries out the distribution commands within each BES. The GL then obtains individual status reports from each group member and sends a group distribution report back to the CCM. The GL is also responsible for notifying the CCM when a member joins or leaves the group.

Advantages of the present invention include provision of a system and method for efficient transmission of data files. The automated system is highly scalable and avoids the unreliability, latency and inefficiencies of implementations heretofore known. Single points of failure are largely eliminated in the method and apparatus according to the invention in which a plurality of group leaders are elected for distributing content to a plurality of back-end content servers. Assignments are created and undertaken in a

manner that facilitates optimal and intelligent distribution or replication of content to different parts of a network, without unnecessarily sending unchanged data.

Similarly, the directed assignment distribution mechanism decreases network load and wasted bandwidth caused by multicasting messages to uninvolved servers. A dynamic tree structure alleviates the administrative costs of manually detecting network server allocations in order to properly address updates.

The content distribution mechanism according to the invention permits highly efficient and robust distribution of assignments and data content from the CCM to each required GL using the store-and-forward tree structure. Dynamic reconfiguration of the content distribution mechanism improves overall system performance by automatically selecting the best available resources to carry out the necessary content distribution tasks. The inventive mechanism is freed from reliance upon any features of IP multicast in Internet routers without sacrificing scalability. The inventive method and apparatus using standard point-to-point communication protocols also avoids potential problems with non-uniform multicast implementations in the WAN. Content distribution via store-and-forward through a dynamic tree structure according to the invention has the advantage of separating the time-critical process of directed content distribution from the bulk of the network overhead generated by dynamic reconfiguration. Grouping remote servers as content targets according to content-type and administrative inputs provides the advantage of eliminating manual configuration and reconfiguration efforts and the occurrence of configuration-related errors in a dynamic network. The ability to carry out the content distribution on standard server hardware, using standard network interface software, permits substantial savings in capital costs, configuration, and maintenance that would be required of specialized hardware.

Furthermore, content distribution management is freed of much of the overhead related to reconfiguration of firewalls at each remote site. Selected message encryption and automated content compression further increase distribution security and efficiency. Scheduler software implemented in the apparatus and method according to the invention reduces unnecessary conflicts in distribution timing. The scheduler also provides significant improvements in synchronization of content received by groups of remote servers. Use of a light-weight, yet robust multicast protocol in the final LAN segment maximizes the efficiency in a web farm where multiple servers can simultaneously receive

the same content updates without having to individually transmit a separate copy to each one sequentially. Back-end reporting to the central content control manager ensures a high degree of synchronization among all targeted servers, network-wide, without requiring that any individual back-end server wait for signals from any other back-end server. Graphical user interface to the content distribution manager simplifies operations by reducing repetitive and error-prone manual steps. The automatic discovery feature of the invention also serves to minimize configuration and management efforts by performing periodic updates to the lists of network segments and their corresponding BESs through communication between the GLs and the CCM. The invention also dovetails with existing performance-oriented products so that service-level reporting can be generated. The content mover can interface with other load-balancing products to provide information about new or removed resources without requiring labor-intensive and error-prone manual reconfigurations. Similarly, the Content Mover can interface with the load-balancing products to instruct the load balancers to remove a BES or a cache from their rotation lists when a BES failed to receive or successfully process and assignment. This allows re-direction of load-balanced requests to only those servers that have the most valid and up-to-date content.

Brief Description Of The Drawings

The foregoing and other features and advantages of the present invention will be more fully understood from the following detailed description of illustrative embodiments, taken in conjunction with the accompanying drawings in which:

Fig. 1 is a block diagrammatic overview of a system architecture according to the invention, in an Internet context;

Fig. 2 is a block diagram of modules that comprise a distribution server in the architecture of Fig. 1;

Fig. 3 is a flow diagrammatic overview of the method of distributing content according to the invention;

Figs. 4A and 4B are diagrams of an illustrative embodiment of a distribution assignment data structure according to the invention;

Fig. 5 is a diagram of an illustrative finite state machine for group leader elections according to the invention; and

Fig. 6 is a system overview of a tree distribution system in an illustrative embodiment of the invention;

Fig. 7 is a diagram for illustration of a store-and-forward distribution tree according to the invention.

Detailed Description

The architecture of the present invention, an illustrative embodiment of which is diagramed in Fig. 1, addresses the need for reliable and efficient distribution of digitized data from a single source to large numbers of dynamically networked recipients. Remote customers or content creators of the distribution method and apparatus use their computers 10 to submit their final content changes to a specified staging server 14. A distribution server 16, having content mover and management functions 18, interacts with servers 26, 28 constituting elected group leaders 30 for distributing content or content changes to distribution agents 32, 34 in each cluster 24. Once the content or change distribution is complete, users can begin to access the updated content from any of the servers 26, 28. User access to content on a given server can be efficiently managed, according to service level agreements, by a completely independent load-balancing method and apparatus, such as are known in the art.

Different types and sizes of digitized information files such as text, photos, videos, voice, or interactive games can be more efficiently managed and more efficiently accessed by content users of the Internet if the content is intelligently placed on a large number of distributed sites. Two distributed sites 24A, 24B are illustrated, each containing a number of file servers 26, 28.

Digital content creators use workstations or other computers 10 to create or digitize information in preparation for publication as "content," such as on the World Wide Web. Content creators may be for example publishers of voluminous periodicals, or real-time information feeds such as stock quotations or auction catalog offerings. The final content, generally in the form of completely new content or changed content, is moved to the staging server 14 by any known and convenient information transfer means, whether automatic or manual, but preferably over a network, such as the Internet 12. The content creators may then test and verify that the content is complete and works properly on the staging server 14 prior to initiating live access by other users in the Internet. Content is then turned over to the centralized Content Control Manager (CCM) 18, running on the distribution server 16.

The Content Manager according to the invention has a distributed architecture. The major components of the architecture are a front-end administrator interface 20, the

Content Control Manager 18, a database 22 for storing policies and configuration information, group leaders 30 for each network segment 24 of recipient backend servers 26, 28, and a distribution agent 32, 34 on each backend server. The front-end administrator interface 20 is a browser based interface effecting the functionality described in detail hereinafter, and accessible to an administrator over the Internet, or directly on a LAN 12c.

The Content Control Manager 18 in this illustrative embodiment is resident on the same physical machine as the staging server 14, however it should be appreciated that it may operate on a separate server (not shown) connected via a network such as a LAN or the Internet 12. It should also be appreciated that any number of machines on the network can be designated as "staging servers" as may be necessary (for example) to service different types of requests, or for different customers, or from different geographic areas.

Administrative inputs for the content mover 18 are obtained from system administrators at the policy management consoles using the administrative interface 20 and stored in the database 22 accessible to the distribution server. Using the management consoles, the system administrators can configure, schedule and monitor various content update jobs and define content groups. The Content Update job is defined as the collection of information that defines what content is to be updated, where it is to be distributed to, when and how often it is distributed, and what policies/rules apply. Content groups are logical groupings of BESs that are serving the same content that will participate in a content update, and are assigned according to file types, ownership, locations, or service levels being offered to subscribers or to users. A content mover assignment comprises all of the necessary instructions to carry out one of the following actions for every relevant backend server: publish, replicate, restore, synchronize, rollback, switch, remove content or publish "hot" content. Designation of "Hot" content is determined through a load-balancer interface whereby the Content Mover is instructed to distribute "flash" (heavily demanded content) to a specified number of BESs. The externally defined load-balancing rules can also be used to trigger content updates, using the content mover, depending upon other externally defined rules determined by the needs of load balancing. In addition, the content mover will handle aborts, file checkpoints, restorations and reconfiguration of content groups of backend servers. The file checkpoints provide a

mechanism to create and manage the version of a complete content roll-out. The invention allows the checkpoints to be saved and restored, as further described below.

The database 22 is a centralized repository for storing configuration information, policies, job information, status logs, snapshots and checkpoints. Examples of information that would be stored in the database include what jobs need to be run, what servers will receive the content update, what options have been selected for encryption and compression and what time a job normally runs. To simplify content synchronization, the database also keeps a history log of each assignment run.

The content control manager (CCM) 40, as illustrated in Fig. 2, is the heart of the content mover system. The CCM in this illustrative embodiment resides on the same machine as the distribution server 16. The CCM oversees the complete process of distributing content from the distribution server to the specified group leaders and monitors further distribution to the BESs. A scheduler 60 notifies the CCM when a job is ready to be run. Scheduled content transactions (such as updates, synchronization, replications, backups, restorations, or rollback) are monitored by the scheduler to avoid server conflicts and to minimize network congestion. Conflicting jobs are queued or cancelled according to administrative inputs to the control database and the console is notified of schedule conflicts. The scheduler is triggered according to events that are stored in the database. The scheduler will, for example, determine whether or not to reschedule the job, if it fails, based on the policy settings in the database.

The Scheduler queries the database looking for assignments that are ready to run. It detects a job that is ready to run by comparing the start time of the assignment to the current time. If the start time is greater than or equal to the current time and has a status of "idle", which means that it is waiting to run, the Scheduler checks to see if any conflicts exists which would preclude the assignment from running properly. If there are no conflicts, the Scheduler sends the assignment to the CCM which, in turn, runs the assignment.

A runtime conflict occurs when a job in progress is distributing to a hosted site on a backend server and a newly detected job needs to distribute to the same hosted site on the same backend server. The Scheduler can detect these conflicts by exploiting the relationship it maintains between content groups, hosted sites, and jobs. A job distributes content to a content group, for example. A content group is comprised of a set of BESs

that belong to the same hosted site. Therefore, there is a relationship between a job, a hosted site, and all the BESs participating in a job.

To detect a runtime conflict, the Scheduler first determines all the hosted sites to which all running jobs are presently distributing. If the hosted site of the pending job is different from those currently listed as busy, there is no conflict. If the hosted site is the same as a busy site, the Scheduler builds a list of all the BESs participating in each running job and compares this list to the list of BESs to which the pending job will distribute. If there are any similar BESs, then the Scheduler has successfully found a conflict. As mentioned earlier, if there are no conflicts, the Scheduler sends the assignment to the CCM which, in turn, runs the assignment.

The CCM then communicates with other management services to obtain information about the task details and policy settings of the job. Job information can be repeated on a scheduled basis without having to modify the task details and policies for each instance of the job. The information will include which servers are to be updated (i.e., "content groups"), the source location of the content to be updated, and the rules, policies and options to be enforced. The source location comprises the directory names to include or exclude, and file types to include in the update. The CCM then determines what files have been changed, how much disk space is required, and the exact commands that each server must execute to perform the changes. The CCM also determines if there are enough backend servers 26, 28 capable of doing the update (a "quorum") and will defer or cancel the update if the required quorum is not met.

Similarly, the CCM handles job requests for content synchronization and replication, and enforces the administrative policies for file compression 46, encryption for transport security 50, and version control 48. Security includes, for example, features necessary to encrypt and compute hash checksums for a block of data. Encryption provides data privacy and hash provides for delivery integrity. The entire assignment file, or an updated file component of an assignment can be hashed and encrypted according to user policies and implemented in the CCM, GLs, and BESs, by means that are known in the art, such as MD5 or SHA1 hash and a symmetric key cipher such as Data Encryption Standard (DES). File compression can be applied to the entire assignment or selected file components, according to user policies, and implemented by standard means.

Version control 48 in the CCM has three aspects: checkpoint, snapshot, and shadow. A snapshot is made of the assignment details during a content update and is used for synchronizing BESs that did not receive the assignment. The "shadow" copy is a complete, exact replica of the content that exists on the live web-server.

A checkpoint copy is defined as a complete 'safe' copy of a site's content at a particular point in time. The creation of a checkpoint copy will be initiated from the management console. Typically, the system administrator will create a checkpoint copy after a major content update has been distributed and verified. The number of checkpoint copies that are maintained by the content distributor services will be a configurable option. The checkpoint copy will be used to restore a site's content back to a previous 'safe' version. The process of restoring the site to a checkpoint copy will be initiated from the management console. For example, the system administrator will select a checkpoint copy to restore and press a button to initiate the restoration. In addition to the checkpoint(s), the Content Control Manager will maintain a snapshot of all the content update assignments that occur after a checkpoint copy is created. The Content Control Manager will automatically create the SnapShot and update the shadow copy at the end of every successful Content update task. A snapshot consists of: content assignment number (assigned by the Content Control Manager), the content job identification number, a time stamp, and the assignment details. A snapshot does not contain the actual file content.

Generally, the method of distributing content according to the invention is illustrated in Fig. 3 which describe a typical "publish" job. After the scheduler awakens the CCM to initiate a task, the CCM retrieves task details 301 and policies from the database. As a further step in content distribution, the CCM determines which content has changed 303, using a delta function described in further detail below. Alternatively, the CCM can automatically detect changes on the designated staging server and distribute the changed content. Depending upon the file changes or other distribution actions necessary, the CCM creates a file 305 of the necessary assignments, as further described below. The CCM then retrieves a current list of Group Leaders 307, and distributes the assignments 309 to the Group Leaders, and from one Group Leader to the next, as determined by the method detailed herein. Group Leaders each ultimately receive assignments destined for members of their own group, and they forward the assignments 311 to the members of their group participating in the update. Each group member receiving an assignment then

interprets the assignment 313 and carries out the actions in the assignment updating its host server. Each group leader then collects the completion status reports 315 from each of the members and relays a consolidated report to the CCM. Each of these steps is discussed in further detail below.

The sequence of events described above and shown in Fig. 3 is for a typical Publish-type assignment. Other assignment types such as Synchronize, Replicate, Restore, and Remove content perform the same sequence of events with certain operational details varying primarily in step 303. For example, a Replicate-type assignment retrieves the task details, policies, name of the host site and the location of the shadow copy from the database 301. As the next step in the replicate process, the CCM invokes the compression module 46 to compress the entire shadow copy contents. The remaining steps for a Replicate-type assignment are the same as shown in Fig. 3.

The CCM will also be awakened by the scheduler to perform jobs such as CreateCheckPoint that do not result in the CCM creating and distributing an assignment file. When the CCM is awakened to perform a CreateCheckPoint job, the CCM retrieves the task details and policies, name of the host site and the location of the shadow copy from the database 301. As the next step in this process, the CCM invokes the compression module 46 to compress the entire shadow copy contents. The CCM then completes this process by invoking the Database API 62 to store the name and location of the CheckPoint file.

The ability to compress files prior to distributing the content will be a configurable option. The UI will provide the ability for the system administrator to indicate what types of files should be compressed. For example, all .HTML file are to be compressed, and all .GIF files will not be compressed. The de-compression of the file content will occur on every backend server during the process of executing the content assignment.

As part of a content update job, the CCM invokes a file list/delta function 42 that determines the files that have changed on the staging server 14 and that need to be pushed out to the appropriate backend servers. Lists of files at a specified directory path on the staging server will be collected and compared against lists of files/directories at a path known as the shadow copy on the distribution server 16. The CCM invokes a versioning function 48 that maintains the shadow copy for every hosted site as described above. Continuing with the delta function, the two lists of files and directories are compared for

differences such as new or deleted file or directory names. Files that are in both lists are then checked to determine when the files were last changed, and a list of changed files is created. The list of changed files is then fed to a delta table builder which identifies the actual bytes changed in the changed file. Any number of file comparison utilities can be used to produce the list of changes to a file, including the standard Unix commands diff and cmp, or WinDiff, or Rsync. Ultimately, a file list/delta table is built, containing a list of changed bytes and their offset locations within the corresponding changed file. This feature results in transmission and distribution of only the changed bytes of data,and offset values. The advantage is a large reduction in transmission time, making the network more efficient.

The CCM contains an assignment creator and manager 44 which responds to scheduler job requests to initiate a distribution job. An "assignment" is a sequence of steps which, when executed, will result in a content transaction being applied on the addressed backend servers. The number of assignments necessary to carry out a job is determined by the size of the content update. An assignment is delivered to a set of GLs and then to a distribution agent or process 32 (Fig. 1) on a backend server where it is parsed to determine what actions to take in furtherance of the job at hand.

To create an assignment, a set of primitives defined by the content mover must be compiled as necessary to carry out the job. For example, to publish a new directory full of files, each destination server's operating system will need to be instructed to create a directory, create each new file name, set the access permissions, and store the contents of each new file from a designated source file. In addition, each file may be compressed or encrypted, or both, either before or after transmission. Similarly, for a content update, a named file must be opened and rewritten with the delta file content, at a specified offset

and hash (checksum) of the corresponding data file, if any. The body 92 of the assignment command buffer contains the set of commands (zero or more) that will be executed in the server, along with the necessary parameters, in the order in which the commands should be executed on the server. The trailer 96 contains a hash of the assignment command sequence.

An assignment data file 84, associated with each assignment command buffer 82 having commands requiring additional data, is shown in Fig. 4B. The assignment data file contains the file content corresponding to each assignment command in the command buffer having a data input argument. As illustrated, the sample data file may contain uncompressed data 96, 98 for multiple destination files, or compressed data 100A, 100B.

As an example of creating an assignment to carry out a content update, the CCM first generates a unique sequence number for the new assignment, and opens a new buffer (or file) to hold the commands and data. The basic commands are largely dictated by the available functionality of the Content Mover. The CCM obtains the file list/delta table and adds commands to create any new directories as necessary. The content itself is then assessed to determine which portions may be non-compressible, compressible, compressible deltas, or non-compressible deltas, and appropriate commands and corresponding data files are added to the buffer. The types of files that are deemed non-compressible are specified by the administrator at the policy management console 20. The CCM then calculates the space that will be required on the server to carry out the assignment. In an illustrative embodiment, an assignment can be up to 4 Gbytes of commands and data. The assignment is then marked as "ready" and queued for delivery by the CCM transport service 80.

The transport service 80 operates in four phases: discovery, global distribution, local distribution, and status reporting. During discovery mode, information is collected from the network elements regarding the connectivity and abilities of the group leaders and their corresponding backend servers. Discovery mode is a continuous process and a given content move is based upon the latest network "discovered". Discovery begins at the network segment level whereby a group leader is elected by the servers in each cluster. The group leaders are then arranged into an appropriate tree structure by the CCM, and all of the backend servers are arranged into their administratively determined content groups.

A content group is a set of backend servers that receives the same content, e.g., by file types, ownership, locations or service level being offered to the subscribers or to the users.

Dynamic system discovery of the content mover system at the remote segment level provides the ability to automatically handle server outages or network reconfigurations. For example, if there are five backend servers in a cluster at a site, one or more could be out of service or occupied for functions unrelated to outward content distribution (e.g. dedicated intranet offline testing). As load-balancing and service level imperatives are periodically addressed (perhaps automatically), the number of available servers at a given site can increase or decrease. The content mover system requires a designation of a group leader at each network segment (i.e., cluster) to handle the store-and-forward to other sites, to handle local distribution of content to the other backend servers, and to provide site reports back to the CCM. A group leader may be pre-assigned for administrative purposes, or it may be dynamically chosen. Such a dynamic designation is carried out automatically through a constantly running process known as "election," detailed as follows.

Each server within a local cluster that is configured to be capable of being a group leader participates in a continuous voting scheme. There may be a list of pre-assigned leaders, or each configured server can be allowed to participate in the voting. The current GL notifies each member in the group with a periodic, non-connection, multicast, User Datagram Protocol (UDP) message, "Leader Alive" (LA), on an agreed "Control Channel," which is distinct from transactions using the agreed "Data Channel." The Control Channel is a Multicast IP and port number combination that all Network segment members listen to. The GL can instruct all members to change their control channel. The Data Channel is a Multicast IP address and port combination that members of Content Group are instructed to listen to. Each GL keeps a list of "Content Group" members, as configured in the CCM. Each Content Group member joins a specific Data Channel (i.e., transmits and receives controls on the Channel). The GL sends assignments to the Data Channel corresponding to its list of Data Channels for a Content Group.

If any server fails to observe theLA messages for a configurable period, then such a server initiates a new election. In its simplest form, the first server to correctly notice the leader is dead and to claim leadership, via an issued "Leader Claim" message, becomes the new leader. If no other server sends a Leader Claim message (LC) to the group within a

preset time, then the vote is over, and the new leader sends its own LA messages to the group. However, each GL candidate may have different priorities, i.e., one may be administratively deemed preferable over another. Where multiple servers determine that the LA messages have stopped, each may attempt to send a Leader Claim (LC) message before receiving an LC message from any other candidate, and a finite state mechanism is required for resolving these conditions deterministically.

This general process can be diagramed as shown in Fig. 5 for the operations defining the five separate states of each participating server: GL known, GL unknown, GL is me, voting open, or conceding to another candidate. Voting is done by sending an LC message, with a priority claim, and at an interval determined by how strong a claim the candidate has on becoming the GL. For example, a recently deposed GL may have the best claim and send the LC message faster than any other candidate could possibly send. This might be the case where a GL was only temporarily too busy to send an LA message, or a server missed three LAs, or the LAN was too busy to allow transmission. An illustrative embodiment uses 5 seconds as the minimum time. Other useful selection criteria include: the number of members the former GL knows about, the candidate's computing, storage, and network resources available at the server, how reliable the server has been, and the amount of recent content a server has acquired (i.e., a newcomer may need to reserve resources for getting its content updated). An initial LC period can also be arbitrarily assigned to be any number larger than the minimum time for the current GL to reclaim an inadvertently lost GL status. In an illustrative embodiment, this value is set to 10 seconds.

As shown in Fig. 5, the normal state of GL known 702 terminates when no LA is heard, and leads to the GL unknown state 703. If an LC is heard before sending an LC, then the concession state 706 is entered, pending an LA from the new leader, at which point GL is known 702. On the other hand, if GL is unknown 703, and the server is first to send an LC, then the state is voting open 705, including a configurable timeout period, for example, in the range of 10 to 15 seconds. Transition from this state depends only on whether another LC is received or not during the timeout. If not, then GL is me 704, and an LA is sent. But if another LC is heard during voting open 705 period, then GL is again unknown 703. Even if the server sends LA, another server could also send LA or LC. Rather than have a battle of LA, the server in GL is me 704 sends an LC and moves back

into voting is open 705 and again waits to hear another LC. Again, if another LC is heard, then GL is unknown 703, and if no LC is heard in the open 705 timeout period, then the machine enters the GL is me state 704 again, and begin sending periodic LA messages.

Once the GL is elected, i.e., there are no more LC messages in the cluster, the new GL expects a new registration message from each member of the group. A server registration message includes a server's IP address, name, and other information. However, given the possible transmission collisions in the LAN segment, some registration messages may not be initially received by the GL. The GL resolves this by multicasting a report on the control channel of all the servers it believes are registered. This report is sent after an initial delay of approximately 3-5 seconds after the last received report. Any server wanting to register but failing to find its name on the multicast GL report, waits a short, randomly determined period and again sends a registration request. In an illustrative embodiment, the interval is comprised of a random element of up to 1 second. This registration message could also have been lost by collision, so after a configurable time out period, the GL retransmits the (updated) report and again waits to hear more registrations. The registration process eventually stabilizes with no new registrations omitted.

The GL now has a complete list of all servers participating in content distribution within that network segment or cluster. The GL reports this information, along with its own IP address, location (e.g., time zone) and other information, such as Network Segment name/id to the CCM. The CCM updates the database used for tracking network status, and updates the list of members that are available to participate in the content groups. The list of GLs and their constituent backend servers is now ready for global distribution of content.

Given a large list of GLs, having different numbers of servers, and located in different places, an essential element for efficient distribution is to create an ordered list of GLs necessary for service to each content group. Thus, one list might quickly move current U.S. stock market data updates, where another list might be used for moving daily wholesale availability updates to catalog servers. Jobs using each type of list can overlap and assignment conflicts can be resolved in the distribution scheduler.

The content mover system adopts a basic tree-structure for global distribution, where a first set of GLs is expected to propagate information to a number of other GLs.

Of course, a simple star is a "tree" in which all GLs are served directly from the CCM, and this may be appropriate where only a small number of GLs are needed for the job. The more interesting problem is where there are literally thousands of servers, spread over hundreds of different far-flung sites, all needing the same instantaneous updates. During the global distribution phase, the CCM delivers the assignments to the group leaders using a reliable point-to-point protocol such as FTP, or Hypertext Transport Protocol (HTTP). Because the TCP/IP protocol stack is part of nearly all popular operating systems, initial development is simplified by using protocols based upon TCP/IP. The group leaders then further distribute the assignments to other group leaders using the store-and-forward method described below.

The CCM will construct a list of the GLs that have reported in the most recent phase of discovery. The GLs will report information to the CCM such as their specific IP address, number of members registered in the group IP addresses of the BESs, free disk space, CPU power, and number of network connections. The list of GLs will be sorted and ordered according to location, performance and distance parameters, as well as speed and reliability of the network connection to the GL. In an illustrative embodiment, the GL list is sorted according to the CPU power (fastest first) and the location, such as time-zone or other physical location indicator. There is a GUI mechanism to allow a system administrator to override (or configure) this behavior and artificially define a static behavior for organization of the distribution sequence. It should be apparent that GLs can be identified by their IP addresses or by another naming convention which maps to the corresponding IP address of the GL, such as "Boston," "London". The system administrator will be able to enter and modify the name of the network segment that the GL is in. If the GL in the network segment changes, the network segment name remains the same.

Store-and-forward requires that GLs share the work in distributing assignments (commands and data) to the more "remote" sites, i.e., further down the tree. As shown in Fig. 6, the CCM is at the root 601 of a distribution tree comprised of a plurality of GLs 603, 605, at varying distances from the root 601. "Distance" is measured as the number of store and forward steps it takes to move an assignment from the CCM to a GL. Each GL at a first defined tree distance from the root, e.g., 603A, 603B, 603C, are required to forward assignments further along the tree to each GL at a further tree distance from the

root, e.g., 605A, 605B, 605C. In a larger tree, additional limbs are added as necessary to reach each GL efficiently.

Since the nature of IP is to send independent packets for routing to their destinations by the best available route, as determined by the routers, certain assumptions can be made regarding the average "distance" at any given time. A viable distance-spanning tree can be derived for efficiently distributing assignments to any combination of remote GLs that supervise the distribution to the targeted backend servers.

The CCM constructs a tree using the sorted list of GLs. Each GL is responsible for distributing an assignment to other GLs first and later to its own members. Assignments are sent from the CCM to a GL or from one GL to another using a reliable point-to-point protocol 609, such as HTTP.

To initiate a distribution, the CCM sends a notification message to the first GL in the list that an assignment is to be obtained from the CCM. A notification message includes the address of a server from which the GL is to obtain a copy of the assignment and the address of the GL to which the notification should be forwarded. Each GL receives notification from the CCM, or from another GL. The GL then gets a copy of the assignment from the specified location (GL or CCM) and contacts the CCM for the notification to forward to the next GL. The GL then sends the notification to the next GL. Once a GL has stored a copy of the assignment (i.e., it was earlier in the CCM list of GLs), the GL can forward copies to other GLs occurring later in the distribution list.

When the CCM transmits a notification, it contains an indication whether the GL is the last on the list. If not the last on the list, the GL requests another GL address from the CCM for forwarding the assignment. The CCM will continue handing out the GL addresses from its sorted list of GLs until all GLs have received the assignment. All GLs that have not already done so will then distribute the assignments to their group members. Communications between each GL and the members of its group is carried out using a combination of a point-to-point protocol 613 and a reliable multicast protocol 611 described below.

Fig. 7 illustrates one example of store-and-forward distribution using a tree, the operation of which will now be explained. In the example, every GL must receive the same assignment, and it would be inefficient to rely upon the CCM to individually contact each GL in seriatim. The transmission power of all available GLs can be used to

dramatically amplify the replication of the assignment issued by the CCM if each GL is given a copy and access to the dynamic list of other GLs to send it to. First, the CCM selects GL1 as the first recipient, and transmits a notification and an assignment to GL1 with an address for GL3. In an illustrative embodiment, the assignment is "requested" by the GL1 after it receives the notification. GL1 forwards the assignment to GL3 and GL1 requests the next GL address from the CCM. While GL1 was busy sending the assignment to GL3, the CCM also sent a copy to GL2, requesting that it be forwarded to GL6. When ready, GL1 requests the next address from the CCM and is told GL4 needs the assignment. GL4 receives the notification from GL1 and requests the copy of the assignment. GL1 copies its assignment to GL4. GL3 has also finished storing the assignment it received from GL1 and also requests the next GL from the CCM. During the creation of the distribution tree, the CCM can quickly determine whether it would be better to have GL2 or GL3 service GL7, and, in the case of this example, decides that GL3 should next forward a copy to GL5 instead. The distribution process continues until all necessary GLs have received a copy of the assignment and each GL has received notice from CCM that no more GLs require a copy (e.g., a null address in the forwarding address of the notification message). Each GL then forwards the assignment to at least some of its servers which make the necessary content interpretations according to the instructions in the assignments.

The mechanism for distribution of assignments from a GL to the back end servers is implemented with a lightweight and reliable multicast protocol. This is referred to as "local distribution" from the GL to the servers. Content assignments are administratively divided into different types of transactions based upon distribution content. The GL uses the Data Channel for distributing assignments and for receiving acknowledgements and reports from the servers. An optional encryption algorithm, as known in the art, can be used in this protocol. It could be used to further increase information security, and to prevent unauthorized systems from participation in the leader election process previously described herein.

A GL determines whether the assignment can be forwarded in one block (e.g., 8 Kbytes), and establishes a temporary "session" for larger assignments. To establish a session, the GL announces to a group that a session is starting, including information about the ID, size, and name of the assignment being sent, as well as the transfer rate and

frame size used. Each member that receives the announcement determines if it needs the assignment and either replies with a completion signal or silently participates in the referenced session. If the GL receives completion signals from all participants, then it knows that all servers have already received the assignment, otherwise it assumes that at least one server needs the assignment. The session information sent by the GL can also be sent in the same message as the periodic "LeaderAlive" messages to the group, in order to further conserve LAN bandwidth.

A group member that has been assigned to participate in a data transfer session will begin to listen for data packets from the GL on the Data Channel. The GL breaks the data into frames of a predetermined size (e.g., 4Kbytes), in accordance with the frame size in the session announcement. Each packet sent by the GL through the multicast UDP contains at least the following information: the Content Assignment ID, a retransmission count, a frame sequence number, and data size. Each receiver checks the frame sequence numbers as each packet is received, and collects a list of missing frame numbers in the received sequence.

After transmission of an entire session, each member participating in the distribution generates and unicasts a status report. The report comprises a list of lost packet numbers, i.e., a Negative Acknowledgement, or NACK. If no packets were lost, then the report contains an empty list, and is interpreted as being a positive acknowledgement for that member. In an illustrative embodiment, the time after transmission for generation of a status report is up to approximately one second. The GL receives the status reports and retransmits the missing frames.

During retransmission of the requested frames, the GL (sender) may retransmit missing frames as each subsequent NACK is received, or it may continue sending data and accumulate a new list of lost frames. The number of NACK messages may be considered as a factor in adjustment of the GL transmission rate. When the GL reaches the end of the data (e.g., a session), it reviews the blocks in the list of NACKs and retransmits only the missing blocks. This delay of retransmission further reduces network traffic and is called NACK aggregation. Alternatively, a NACK can be sent after a predetermined time, rather than waiting for transmission of an entire session.

Each block transmission includes a retransmission count generated by the GL, initially set to one, and incremented for each retransmission in response to further NACKs.

The retransmission or pass number is used by a Group Member to find lost frames. The GL continues to retransmit blocks in response to NACKs until it gets the status reports from all or a quorum number of all participating BESs. This signifies that each participant has received each block, up to and including the final block. Once a group member has received a complete session of data, i.e., an assignment, the assignment is passed to the local content interpreter.

The content interpreter running in each back-end server, parses the received assignment and interacts with the BES operating system to carry out the instructions of each assignment on each addressed backend server. For example, the interpretive functions can be carried out in a content interpreter in a group member running the CI function, and the resulting commands can then be communicated to another group member by communication means known in the art, such as terminal emulation, or file transfer of a command file for execution by the group member. The content interpreter converts the assignment instructions into commands and corresponding parameters and sends them to the operating system of the server. Examples of standard commands are "create a directory with the following path name and permissions," and "create a file with the following compressed contents." The content interpreter then interprets the response codes from the operating system and creates reports of assignment completion. Once an assignment is completed, the content interpreter sends its report to the Group Leader.

Each registered group member participating in a content move must send a job status report for each assignment it receives. The report may be sent either using user Datagram Protocol UDP or Transmission Control Protocol TCP. The Group Leader sends a unicast request to any participating group member who did not report or whose response was lost. If the number of missing reports exceeds a configurable number such as half the number of group members, then the GL sends a multicast request on the LAN, including a list of group members. If there are fewer servers that have not reported than have reported, the GL sends the list of those whose reports are missing. Otherwise, it sends the list of those whose reports were received. In either case, each server interprets the request and the list and retransmits its own report, if necessary. The GL retransmits its request and corresponding list until each participant has reported or a configurable maximum number of transmission has been reached. Each member transmits its report at a time determined as a random multiple of a predetermined interval on the order of a few

seconds. This avoids "report implosion" in which all servers would attempt to reply at the same instant, causing massive transmission collisions and retransmission timeout delays, in which case the reporting process would take much longer to be completed.

When the number of lost frames does not decrease after a configurable number of tries (e.g., ten tries), and the GL does not have a quorum number of reports, then the GL reports a problem to the CCM. Otherwise, the GL forwards a comprehensive assignment completion report to the CCM. The report contains the complete list of servers that have reported successful completion of the assignment. In either case, transmission from GL to CCM is accomplished by using HTTP protocol. The CCM processes each GL report for each assignment, updates its database with assignment status, and optionally logs the reports.

As a final step in carrying out a content distribution job, the CCM synchronize content switchover in all back end servers by sending out another assignment after the content has been successfully distributed. This "commit" assignment is distributed to the GLs and the servers by the same mechanism as any other assignment. Upon receipt, each content interpreter performs the necessary steps as triggered in the commit assignment, such as by swapping directory pointers between the current "live" content and the newly updated content. In an illustrative embodiment, switchover is accommodated in the following way. The directory structure in the "live" directory is replicated in a "temp" directory. Each command in a received assignment causes corresponding directories to be added or removed in the temp directory, and new or modified files added. For any commands to remove a file, an empty file with the deleted filename is added to temp. Upon receipt of the "commit" assignment, a "diff" function copies each file from "live" to "temp" that was not already in "temp" and a list of moved files is saved. The web server is then reconfigured to point to the "temp" directory (e.g., using ISAPI redirection, NSAPI or Apache re-run config), the "live" directory is renamed as "backup", and the "temp" directory is renamed as "live."

Similarly, for a "rollback" assignment, for each file in the list saved during the previous live directory creation, move the file from "temp" to "backup", switch the server to point to "backup" directory, and remove the "live" directory by renaming "backup" directory to "live."

Each server then sends its report to the GL, the GLs send their own reports to the CCM, and the commit status reports are processed and logged by the CCM. The GL Assignment Status Report includes the list of servers that have reported successful completion of the assignment. Another type of reports that the GL sends to the CCM. A Network Segment (NS) Report is sent whenever the GL detects a change to a BES, e.g., when a BES went offline. The CCM updates the database for every report received. The GUI will periodically poll the database and refresh its screens with updated status information.

When a new BES is added to a network segment, the GL will send CCM report. CCM will update the database list of computers. The browser-based User Interface will show the new computer by displaying a new computer icon for this entry. The administrator then needs to add this computer to one or more content groups. The scheduler will note the change, and trigger a Replicate job to be sent to the CCM.

The interface between the Content Mover and external load-balancing systems can be beneficially exploited for reducing distribution overhead, while keeping content "fresh" at essential sites. As illustrated in Fig. 6, the CCM 601 and/or GL 603 will notify any load balancer 615 or virtual resource management (VRM) device when it needs to reduce the load on the group leader for a content assignment. The load balancer or VRM may also be notified to add or remove any of its network resources (603, 605, 607) from rotation while an update is in progress. The network resources may be added or removed in groups or one at a time. Some examples of why this may occur include the case wherein the CCM or GL determined that content on a particular network device is stale, response time for a particular network device is slower than others, or a new device is being added to the content group.

The CCM and/or GL will maintain a list of files currently contained in each network cache 617 on each network segment (e.g., LA, London, Paris, Boston). When a content update occurs, the list of files contained in the cache will be compared, and new content will be automatically distributed to the network cache. This guarantees that content being served from network caches is always up to date and fresh. The update to the cache can be a scheduled update or it can happen automatically. In addition, the CCM and/or GL will send invalidation messages to the cache on each network segment. For

third-party compatibility, these messages need not be in the form of a proprietary assignment command, and would be created to conform with third-party specifications.

As a further example of an external interface with Content Mover, the CCM and/or GL will contain replicated information of a search engine and will automatically update the search engine with fresh content, i.e., according to a schedule or when content changes.

Although a distribution server and CCM have been described as being co-resident in a server host, the architecture of the content mover does not preclude one from having multiple CCMs and multiple Distribution Servers operating simultaneously. One application of this alternative embodiment would be for geographical partitioning of management whereby each CCM/distribution server would manage its own GL distribution list. Furthermore, most (if not all) of the individual components illustrated as being implemented within the distribution server 18, in Fig. 2, such as the GUI 64, logger 66, scheduler 60, database manager 68, and database 22, can each be implemented in a distributed fashion across multiple computers connected via a communication network.

Similarly, although the invention describes distribution of content to a BES in a distributed network, it should be noted that the Content Mover can distribute data of any type to caches or any other network device on any distributed network or wireless network, including, but not limited to satellite.

Although the group leader has been described in an illustrative embodiment as a remote element in the system such as a separate computer server 26A, it will be appreciated that in an alternative embodiment, a group leader may also be hosted on the same computer hosting the distribution agent 34 in Fig. 1, and distribution server 16. This alternative arrangement facilitates use of a GL to serve a local cluster of servers 24A. It should be apparent that this alternative will also require corresponding changes to simplify the protocols used to exchange data between the distribution server and the GL. Similarly, a group leader may share hosting with a backend server's distribution agent (and its content interpreter). This would be convenient where the host server is well adapted to be a content server as well as a group leader, and also permits better dynamic load balancing among all hosts in the web farm cluster, regardless of which server is currently operating as the GL.

While automated processes are described herein for configuration of clusters, it should be appreciated that in an alternative embodiment, the selection of a Group Leader, identification of live servers, and allocation of live servers to content groups can all be done manually, or using a combination of existing tools and network utilities known in the art. For example, a single Internet Control Message Protocol, ICMP "ping" (echo) message and response may be sufficient to determine which servers are live, and a CCM script with a list of IP addresses of every possible server could slavishly ping each server, resulting in a list of live servers. Similarly, network distances can be roughly measured using ping, or more sophisticated performance measurement tools for end-to-end delay.

It will be appreciated by one skilled in the art that a content interpreter (CI) can be customized to interpret assignment commands into compatible commands for the particular operating system running on any BES. This offers enhanced portability as the content mover can handle distribution to any kind of host computer for which an appropriate CI has been crafted.

It will be appreciated by one skilled in the art that although the content mover has been described for distribution of passive file contents, it will also be useful in mass distribution of application files (executables), configuration information, or registry settings necessary to install new executables onto a BES, a cache, or any other network or wireless device on a distributed network. Similarly, the invention can be adapted for distribution of configuration updates for existing applications. The Assignment Creator and content interpreters would be modified as necessary to implement such additional commands to the BES, and to obtain the requisite status reports and log them appropriately.

Although the invention has been shown and described with respect to illustrative embodiments thereof, various other changes, omissions and additions in the form and detail thereof may be made therein without departing from the spirit and scope of the invention.

What is claimed is:

1. A system for distributing information to a plurality of group members connected via a communication network, comprising:

a content control manager (CCM) for processing said information into at least one assignment for a distribution job, and for managing said distribution of said at least one assignment to said plurality of group members;

a set of group leaders, each having a corresponding set of group members and each communicating with said CCM, for forwarding said assignments to other group leaders in said set of group leaders according to commands from said CCM, and for forwarding said assignments to said corresponding set of group members belonging to a set of destinations;

each of said group members being associated with a command interpreter for carrying out said assignments on each group member,

whereby said information is communicated to each of said group members within said set of destinations.

2. The system of claim 1 further comprising a database including:

a copy of said information to be distributed;

a set of content groups each comprising a set of destinations;

a list of group members currently included in at least one of said set of destinations; and

a set of user-specified policies relating to distribution of said information, including a set of scheduling parameters.

3. The system of claim 2 in which said database is further accessible from a workstation on said network, said workstation having a graphical user interface adapted for interactive operation of said database by an operator.

4. The system of claim 1 in which said distribution server further comprises

a scheduler for scheduling a distribution job according to events scheduled in a database,

a communication interface to said network for communicating with at least one of said set of group leaders; and

a graphical user interface to a database.

5. The system of claim 1 in which said distribution server further comprises

a first version of a source file;

a second version of a source file; and

a file delta generator for comparing said first version with said second version to generate a set of changes between said first and second version;

whereby a set of information for distribution comprises said set of changes.

6. The system of claim 1 in which said at least one assignment is further comprised of:

a set of commands for at least one of said command interpreters; and

if said assignment is for distribution of information, further including a set of said information for distribution.

7. The system of claim 6 in which said assignment further comprises

a header having:

an assignment identifier,

an assignment type selected from a set of assignment types, and

an error detection hash for said set of information for distribution.

8. The system of claim 1 in which said content control manager further comprises

a distribution manager for managing distribution of said at least one assignment including

a list of group leaders corresponding to said set of destinations;

a transmission process for sending a copy of said at least one assignment to each of said group leaders on said list as a set of data packets using a reliable transport protocol;

a completion process for determining when a copy of said assignment has been successfully sent to each of said group leaders on said list; and

a verification process for collecting reports from each group leader and for determining which destination of said set of destinations has received said assignment and successfully carried out said assignment.

9. The system of claim 1 in which each of said group leaders further comprises

a first reliable transport protocol for communicating with said CCM and for communicating with other group leaders;

a second reliable transport protocol for communicating with each of said group members corresponding to said group leader;

a store-and-forward process for receiving an assignment from said CCM, for receiving an address of another group leader of said set of group leaders from said CCM, for sending a copy of said assignment to said other group leader if so commanded by said CCM, and for requesting additional group leader addresses from said CCM until said CCM signals that each of said set of group leaders has received a copy of said assignment;

a distribution process for forwarding a copy of said assignment to each of said corresponding group members and for verifying that each of said group members has successfully handled any commands and information of said assignment, and for generating and sending a report to said CCM according to said handling by each of said group members.

10. The system of claim 9 in which

each of said corresponding group members is connected to said group leader by way of a network supporting a multicast protocol; and

said second reliable transport protocol is a multicast protocol,

whereby each group member in said set of destinations receives a copy of said assignment at substantially the same time as each other group member in said set of destinations.

11. system of claim 1 in which each of said set of group leaders communicates with its said corresponding set of group members using a reliable multicast protocol, whereby each group member in said set of destinations receives a copy of said assignment at substantially the same time as each other group member in said set of destinations.

12. The system of claim 1 in which each of said group members further comprises

a reliable multicast transport protocol for receiving a copy of said assignment from said corresponding group leader;

a reliable transport protocol for communicating with said corresponding group leader; and

a reporting mechanism for generating and sending a status report to said group leader after receiving and processing said assignment.

13. The system of claim 1 in which at least one of said set of group leaders communicates with said CCM and with other group leaders of said set via TCP/IP over the Internet and at least one of said group members receives said assignment from said corresponding group leader using a reliable multicast protocol over a local area network.

14. A method of distributing information via a communication network, comprising the steps of:

determining a content change in a source file;

determining a set of destination servers for receiving an update;

generating a sequence of update commands for said destination servers;

obtaining a list of clusters for delivery of said sequence to said set of destination servers;

communicating said sequence to each of said clusters on said list;

forwarding said sequence to each destination server within said cluster; and

executing said sequence on at least some of said destination servers.

15. The method of claim 14 further comprising the steps of:

monitoring within each said cluster a status indication of which destination servers properly execute said sequence of commands; and

reporting said monitored status to a designated monitoring center.

16. The method of claim 14 in which said step of determining a content change further comprises the steps of:

generating a list of filenames that occur in both said current version and said present version;

comparing each file, corresponding to each filename in said list, between said prior version and said current version to build a table of file changes according to each filename;

generating a set of directory names and filenames that occur in either said prior version or said current version, but not both; and

reporting said set and said table as a content update.

17. The method of claim 14 in which said step of generating a sequence of update commands further comprises the steps of:

selecting a command set according to a type of destination server for each destination;

identifying any data change in said determination of content change that requires transmission of file data; and

making a list of commands corresponding to each said determined content change, along with associated references to file data for each identified data change;

whereby said list of commands and said associated references are sequenced in a data structure for transmission to each destination server.

18. The method of claim 17 in which said step of making a list of commands further comprises the steps of:

evaluating configuration input to determine which file data must be encrypted or compressed, or both, prior to transmission;

carrying out such encryption or compression, or both, as indicated for said associated file references; and

inserting corresponding decryption or decompression commands, or both, into said list of commands.

19. The method of claim 14 further comprising the steps of:

administratively dividing said set of destination servers into a set of content groups;

determining a cluster identifier corresponding to each destination server; and

nominating a group leader as a communication channel for each said cluster identifier;

whereby a list of clusters is created for each content group and a nominated group leader represents each cluster.

20. The method of claim 14 in which said step of communicating said sequence further comprises the steps of:

(A) nominating a group leader for each said cluster;

(B) ordering said list of group leaders for carrying out a store-and-forward distribution;

(C) sending a notification to a target group leader to obtain a copy of said sequence from a named source;

(D) receiving said notification at said target group leader and requesting said sequence from said named source;

(E) receiving said sequence from said named source and reporting completion of said receiving;

(F) responding to said report by notifying a next target leader determined from said ordered list to obtain a copy of said sequence from a named source; and

(G) repeating steps (C) through (G) for each group leader until said list of group leaders is exhausted.

21. The method of claim 14 in which said step of determining a set of destination servers for receiving an update further comprises the step of identifying destination servers having stale content, wherein any of said set of destination servers were omitted from one or more earlier updates.

22. The method of claim 14 in which said step of determining a set of destination servers for receiving an update further comprises the steps of:

identifying a set of network resources for which load must be reduced; and

notifying a resource manager of said set of network resources to be taken out of service during an update.

23. The method of claim 14 in which said step of determining a set of destination servers for receiving an update further comprises the step of:

managing a load factor on a set of network resources by communicating with a resource manager to determine which of said network resources should be added or removed from service during a particular update;

whereby selected network resources can be taken out of service to receive an update, thereby reducing the load on said network resources.

24. The method of claim 23 in which said network resources are selected from the set of group leader, destination server, and group of destination servers.

25. A method for selecting a group leader among servers in a multicast network segment comprising the steps of:

configuring a set of said servers to participate in electing a leader, each said server having a corresponding voting priority;

determining when a new leader is needed; and

electing one server of said set to become said new leader.

26. The method of claim 25 in which said step of configuring a set of servers further comprises the steps of:

measuring a set of leader selection parameters in each participant in said set; and

calculating the corresponding voting priority according to said measurements.

27. The method of claim 25 in which said step of determining when a new leader is needed further comprises the steps of:

configuring each server that is not currently the group leader to listen for periodic messages from said group leader;

adapting each server to send said periodic messages only if said server is currently the group leader;

waiting a configurable period after no periodic messages are heard; and

multicasting said voting priority to each participant.

28. The method of claim 27 in which said periodic messages are multicast on a predetermined network channel comprised of an IP multicast address and a port number.

29. The method of claim 27 in which said multicasting is addressed to a preconfigured IP multicast IP address and port combination for each server of said set of participating servers.

30. The method of claim 25 in which said step of electing further comprises the steps of:

sending a claim of leadership containing a sent voting priority;

listening for other servers to claim leadership;

comparing a received priority in any other claims to leadership with said sent voting priority; and

determining said new leader according to the server having claimed leadership with the highest voting priority.

31. The method of claim 30 in which said step of sending a claim of leadership is implemented using a multicast message on said multicast network segment.

32. A system for determining a group leader among a group of servers comprising:

a set of participant servers including at least some servers capable of participating in electing a group leader;

a communication channel from each participant to each other participant;

a monitor process in each participant to determine which server is the current group leader; and

an election process in each participant to calculate a voting priority of said participant and to select a new group leader according to said voting priority, said election process triggered by said monitor process.

33. The system of claim 32 in which said monitor process further comprises:

a listener in each participant for determining how long since a group leader alive message has been heard on said communication channel;

a transmitter in each participant, operable in an elected group leader, that periodically signals each other participant who the current group leader is; and

a trigger adapted to detect that a group leader has not been heard from for a time longer than a threshold time, according to the period of said periodic signal.

34. The system of claim 33 in which said threshold time is configured such that a trigger will occur no less than five seconds after the last group leader alive message was received by said monitor process.

35. The system of claim 32 in which said election process further comprises a state machine adapted to transition from a temporary state of group leader unknown to a stable state of either group leader known or group leader is me, according to the following steps:

in said state of group leader unknown, if one or more group leader claim messages (LC) are received in which a received voting priority is greater than said calculated voting priority, then transition to a concession state and wait for a group leader alive message (LA); and if no LC is received before a period determined by said calculated voting priority, or no received voting priority is greater than said calculated voting priority, then transmit an LC including said calculated voting priority, and transition to a voting open state;

in said concession state, when an LA is received, transition to said group leader known state;

in said voting open state, if no LC or LA is received for a predetermined time interval, then transition to said group leader is me state and transmit an LA; and if one or more LC is received prior to said predetermined voting time interval, then transition to said group leader unknown state and transmit an LC; and if an LA is received, then transition to said group leader known state;

in said group leader is me state, periodically send an LA until an LC or LA is received, and then transition to said voting open state and send an LC; and

in said group leader known state, upon a trigger from said monitor process, transition to said group leader unknown state and transmit an LC;

whereby a participant having the highest calculated voting priority is elected group leader.

36. The system of claim 33 in which said predetermined voting time interval is no greater than 15 seconds.

37. The system of claim 32 in which said voting priority for each participant is determined dynamically according to at least one parameter selected from the set of: how recently was said participant the group leader, the number of servers known to said participant, the amount of resources available to such participant, reliability of the participant, the amount of recent information content the participant has acquired, and a user-specified priority factor.

38. A method for determining registration of members of a cluster of servers on a network segment comprising the steps of:

(A) designating a group leader on said network segment;

(B) each member sending a registration message to said group leader;

(C) said group leader multicasting a registration report including an identifier corresponding to each registered member;

(D) sending another registration message from any member receiving said registration report in which said member's corresponding identifier is missing;

(E) repeating steps (C) and (D) until each said member receives a registration report including its own corresponding identifier as a registered member.

39. The method of claim 38 in which said step of designating said group leader is carried out among a set of servers of said cluster according to a voting priority determined by at least some of said members from a set of dynamic parameters measured within themselves.

40. The method of claim 38 in which said step of multicasting said registration report occurs after a configurable interval has expired since the most recent registration request was received by said group leader.

41. The method of claim 38 in which said step (D) of sending another registration message occurs after a preconfigured interval, after receipt of a registration report or a registration request, comprised of a fixed interval and a random interval, said random interval being up to one second.

42. The method of claim 38 in which said registration requests include at least some identification information selected from the set of: a server's IP address, a server's name, a server's port number, and a secret key.

43. The method of claim 38 further comprising the steps of:

processing said registration reports in said group leader to create a cluster report; and

transmitting said cluster report to a network distribution server, whereby said cluster reports are dynamically collected from all clusters in said network.

44. The method of claim 43 in which said cluster report further includes at least some group leader information selected from the set of: a list of registered members, identification information from at least some of said registered members, a network segment identifier, an IP address of said group leader, and a location parameter for said group leader.

45. A system for distributing information to a set of destination nodes connected via a communication network comprising:

a reporting process in each destination node for generating and transmitting a report to a distribution manager, said report containing an identification of said destination node and corresponding destination node parameters, whereby said destination node offers to become a participant in a distribution job;

said distribution manager connected to said network and configured to receive said reports from said destination nodes and to create a prioritized list of destination nodes selected as participants in a distribution job according to said destination node parameters, and having a management process for sending information to each participant;

said management process adapted to send each participant instructions to obtain a copy of said information either from said distribution manager or from another identified participant, until each participant has received a copy of said information;

each participant having a store-and-forward process configured to receive instructions from a prior participant or from said distribution manager and to request a copy of said information from said prior participant or from said distribution manager, and to thereafter request further distribution instructions from said distribution manager until instructed that no other participants require said information;

whereby each participant obtains a copy of the information and the distribution manager obtains confirmation that each destination node has obtained said information.

46. The system of claim 45 in which the destination node parameters, by which said prioritized list is created, are selected from the set of parameters consisting of location, destination node performance, distance from other destination nodes, transmission speed of said destination node's network connection, and reliability of said network connection.

47. A method of distributing information to a set of destination nodes connected via a communication network comprising the steps of:

obtaining a list of destination nodes desiring to participate in a distribution;

prioritizing said list according to parameters associated with each destination node on said list;

issuing instructions to each destination node according to the prioritized list order, said instructions including the identification of a source for obtaining said information and an identification of the next destination node on the prioritized list;

distributing said information according to said instructions; and

notifying each destination node when the prioritized list is exhausted.

48. The method of claim 47 in which said step of prioritizing further comprises the steps of:

obtaining a set of parameters corresponding to said list of destination nodes; and

sorting said list according to at least some of said set of parameters;

whereby the sorted list provides a prioritized ordering of destination nodes to receive said information being distributed.

49. The method of claim 47 in which said steps of issuing instructions and distributing said information further comprise the steps of:

(A) obtaining an address of a target node on said prioritized list;

(B) sending a notification message to said target node, said notification containing the address of a source node having an information file to distribute;

(C) receiving said notification message at said first destination node and requesting a copy of said information from said identified source node;

(D) reporting when said target node has successfully received said copy of said information;

(E) responding to said report by selecting a next target node address on said prioritized list and repeating steps (A) through (E) until said prioritized list is exhausted;

whereby said copy of said information is sent to each target node on said list in an order determined according to the order of the list for each successful copy.

50. The method of claim 49 in which said steps of sending and receiving are carried out using a reliable transport protocol across said communication network.

51. The method of claim 47 in which said step of obtaining a list of destination nodes further comprises the steps of:

electing a group leader among a set of destination nodes on a network segment;

sending reports of the respective address and capabilities of each destination node to said group leader;

collecting and processing said reports in said group leader;

generating a network segment discovery report in said group leader;

sending said network segment discovery report to a distribution manager; and

arranging said list of destination nodes according to at least one of said network segment discovery reports.

52. The method of claim 51 in which said distribution server further comprises a database for storing said reports and a graphical user interface for providing an interface for system administration personnel.

53. The method of claim 51 in which said step of arranging said list further comprises the steps of:

building a database of reports collected from said group leaders;

collecting distribution policies;

administratively dividing said destination nodes into content groups;

collecting information about the network communication delays;

creating a list of group leaders corresponding to each content group; and

sorting each said list of group leaders according to at least one of said reported destination node capabilities, distribution policies, and information about the network communication delays.

54. A system for distributing information to a set of destination nodes connected via a communication network comprising:

a discovery process for discovering destination nodes in a network;

a distribution server for receiving the results of said discovery process, comprising

at least one network group comprised of a set of destination nodes;

a list of destination nodes comprising at least one user-defined content group;

a user interface for obtaining distribution policies; and

a distribution process for managing the transmission of said information to each of said network groups.

55. The system of claim 54 in which said discovery process further comprises:

a group leader for collecting reports from at least one destination node in a network segment and for processing said reports into a network segment report and for sending said report to said distribution server.

56. The system of claim 54 in which said network group is created through the process of receiving reports from a set of group leaders, in which at least some reports identify destination nodes under the control of the corresponding group leader.

57. The system of claim 54 in which said distribution process further comprises a mechanism for:

arranging a list of network groups according to at least one parameter selected from the set of: physical location, transmission delay for reaching said network group from said distribution server, and a measure of destination node capabilities of at least one destination node within each network group; and

transmitting said information to each network group according to said arranged list.

58. The system of claim 57 in which said system further comprises:

a report generator for generating a report from any destination node that has changed its availability status or system capabilities, or has successfully received an information transmission from said distribution process; and

a report transmitter, in a group leader for each cluster of destination nodes, for collecting reports from any destination node in the corresponding cluster, and for processing said reports and transmitting a consolidated report to the distribution server.

59. The system of claim 54 in which said discovery process further comprises:

at least one network cluster comprised of destination node connected to each other with a multicast communication medium;

an election mechanism for dynamically selecting one destination node of each said at least one network cluster to be group leader; and

a reporting process in which each said group leader solicits reports from each destination node in the corresponding cluster, creates a profile report for the cluster, and transmits it to said distribution server.

60. The system of claim 59 in which said reporting process is triggered by an event selected from the set of: a timer, a group leader receiving a registration request from a destination node in said network cluster, a change of group leader, a user input, an information transfer from said distribution server, and an instruction from said distribution server.

61. The system of claim 54 in which said distribution process further comprises:

a list of destinations, each corresponding to a network group, said list adapted to represent a transmission tree structure in which each destination on said list becomes an information source for any later destination occurring later in said list;

a point-to-point transmission process for sending information from one of said information sources to any of said later destinations;

a distribution manager containing an information source, and for obtaining said list of destinations and requesting a point-to-point transmission of said information to each destination according to said list, and for determining when transmission has been completed for each destination.

62. The system of claim 61 in which said distribution manager further comprises:

a process for creating an information notification message for transmission, said notification including an earlier destination address from said list and an information identifier, and for triggering transmission to said next destination;

whereby said next destination receives a notification that information corresponding to said information identifier is to be obtained from one of said information sources.

63. The system of claim 61 in which said list is ordered according to parameters selected from the set of: the CPU power of the destination, the network location of each destination, and user inputs.

64. The system according to claim 54 in which each network group further comprises:

a group leader for receiving information from an information source and for instructing other group leaders to request information from an information source;

a local distribution mechanism in which a group leader transmits said information to each destination node in its corresponding network group using a multicast protocol; and

an interpreter in each said destination node for executing a set of commands in said information, and for reporting completion status of said execution.

65. The system according to claim 54 in which said distribution process is responsive to said distribution policies and identifies which network groups correspond to any destination made in said content group.

66. A method for determining the completion status of processing by individual nodes of a group of destination nodes on a multicast data channel, said group having a group leader, comprising the steps of:

notifying an assignment processor in at least one of said destination nodes that an assignment has been received for processing;

determining completion status of said processing by at least one of said destination nodes; and

notifying said group leader of said completion status by a message from said at least one destination node such that the group leader acquires a collective indication of status from nodes within said group.

67. The method of claim 66 further comprising the steps of:

determining whether the number of non-reporting nodes in the group exceeds a configured number or proportion of nodes;

and transmitting a multicast request from said group leader (GL) to nodes in said group where said request is formatted according to said number of non-reporting nodes, and otherwise transmitting a request to each non-reporting node; and

repeating said determination of the number of non-reporting nodes and transmitting a multicast or unicast request until either each node has reported its status or a preconfigured number of requests has been transmitted.

68. The method of claim 67 further comprising the steps of:

when the number of non-reporting nodes is less than the number of reporting nodes, listing said non-reporting nodes in said multicast request;

when the number of non-reporting nodes is not less than the number of reporting nodes , listing said reporting nodes in said multicast request;

interpreting said multicast request in each node to determine whether any given node in the group should transmit a missing report to said GL; and

transmitting said missing report according to said determination, after a time delay configurable for each node.

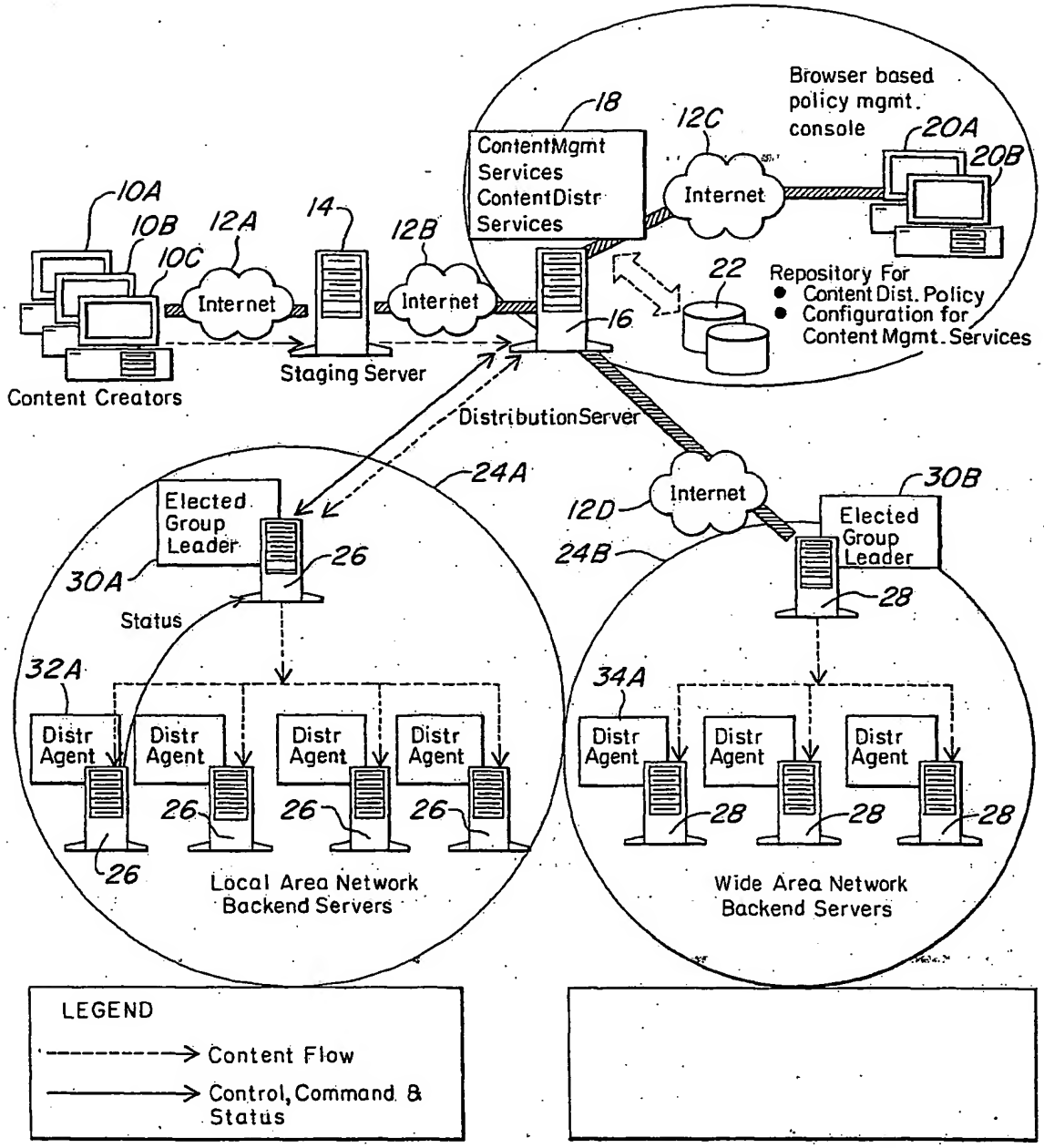
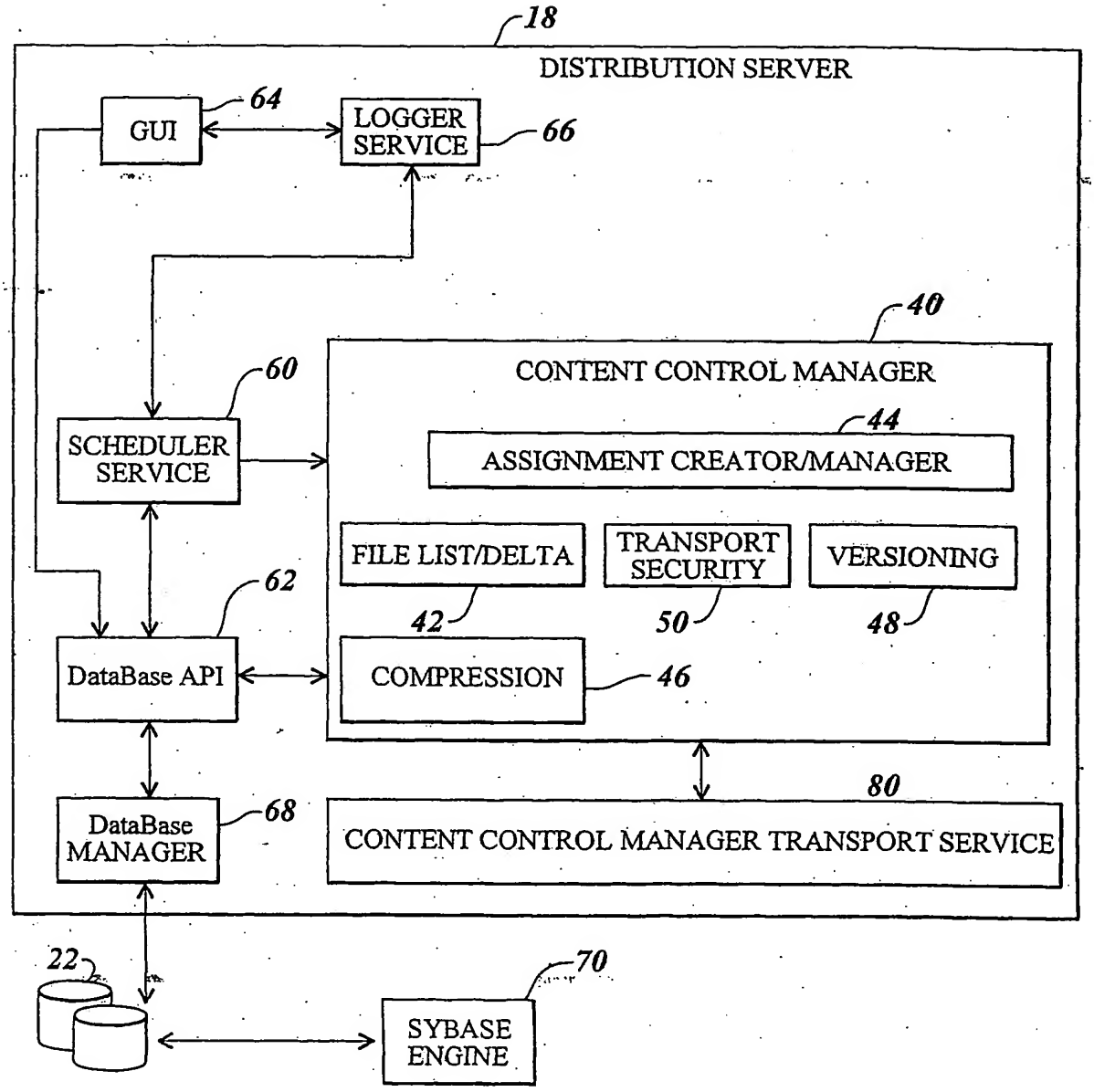


FIG. 1



REPOSITORY FOR:

- CONTENT DIST. POLICY
- CONFIGURATION FOR CONTENT MGMT. SERVICES

FIG. 2

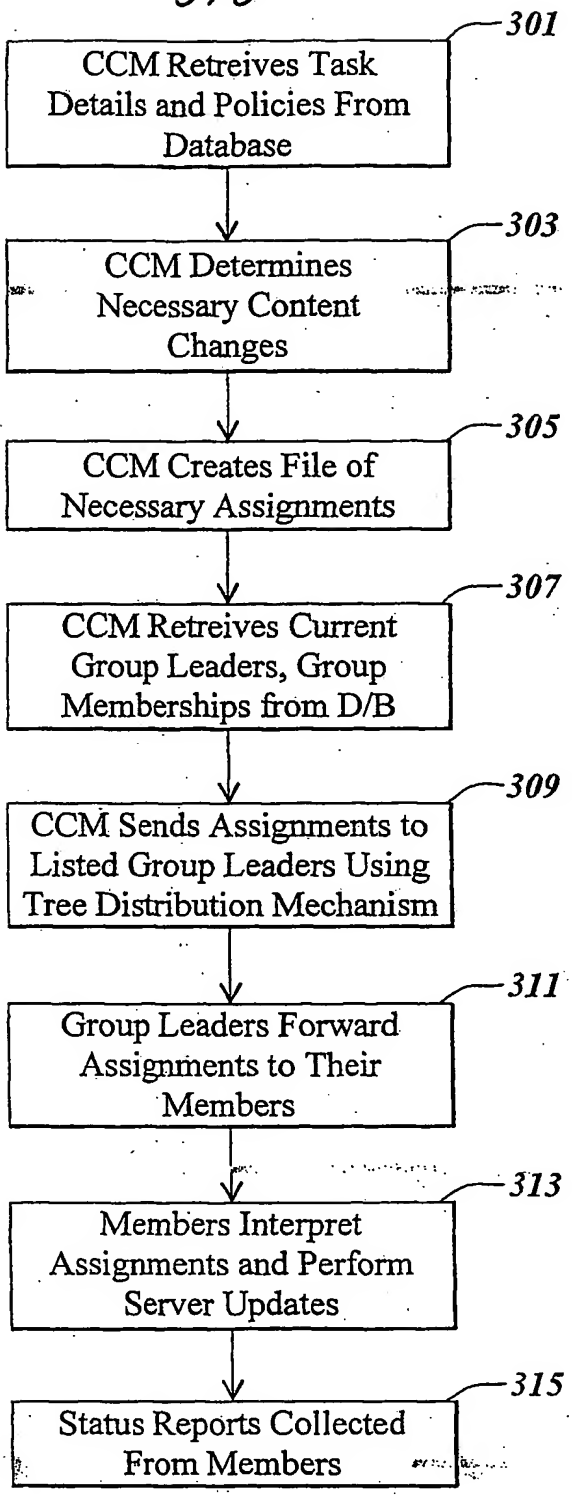


FIG. 3

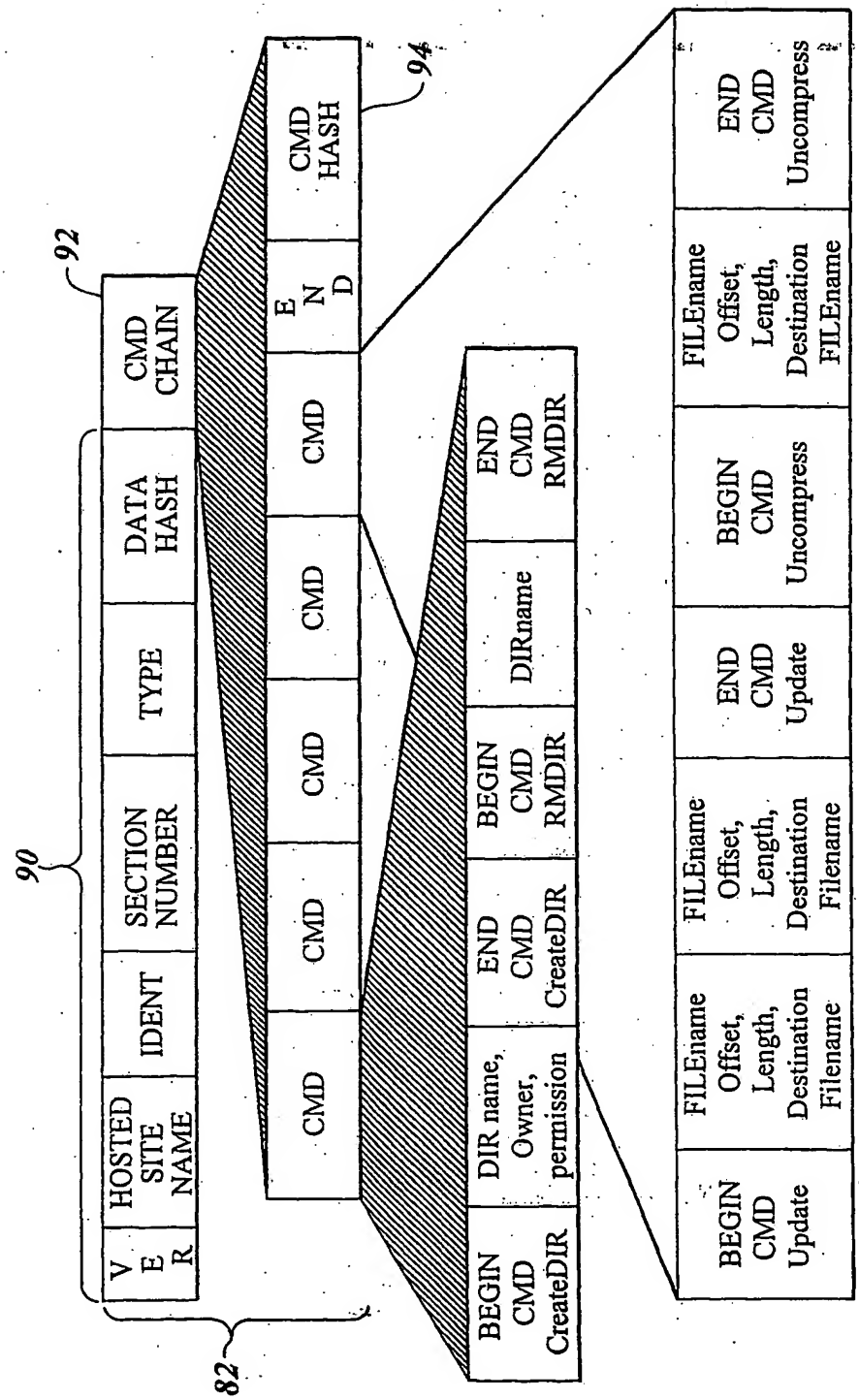


FIG. 4A

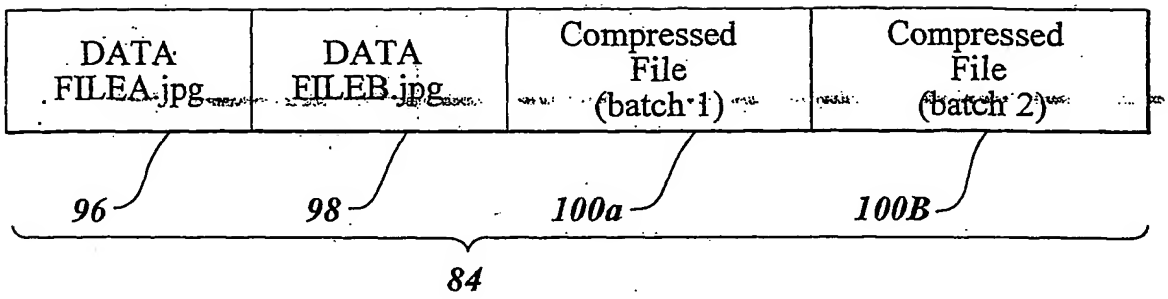


FIG. 4B

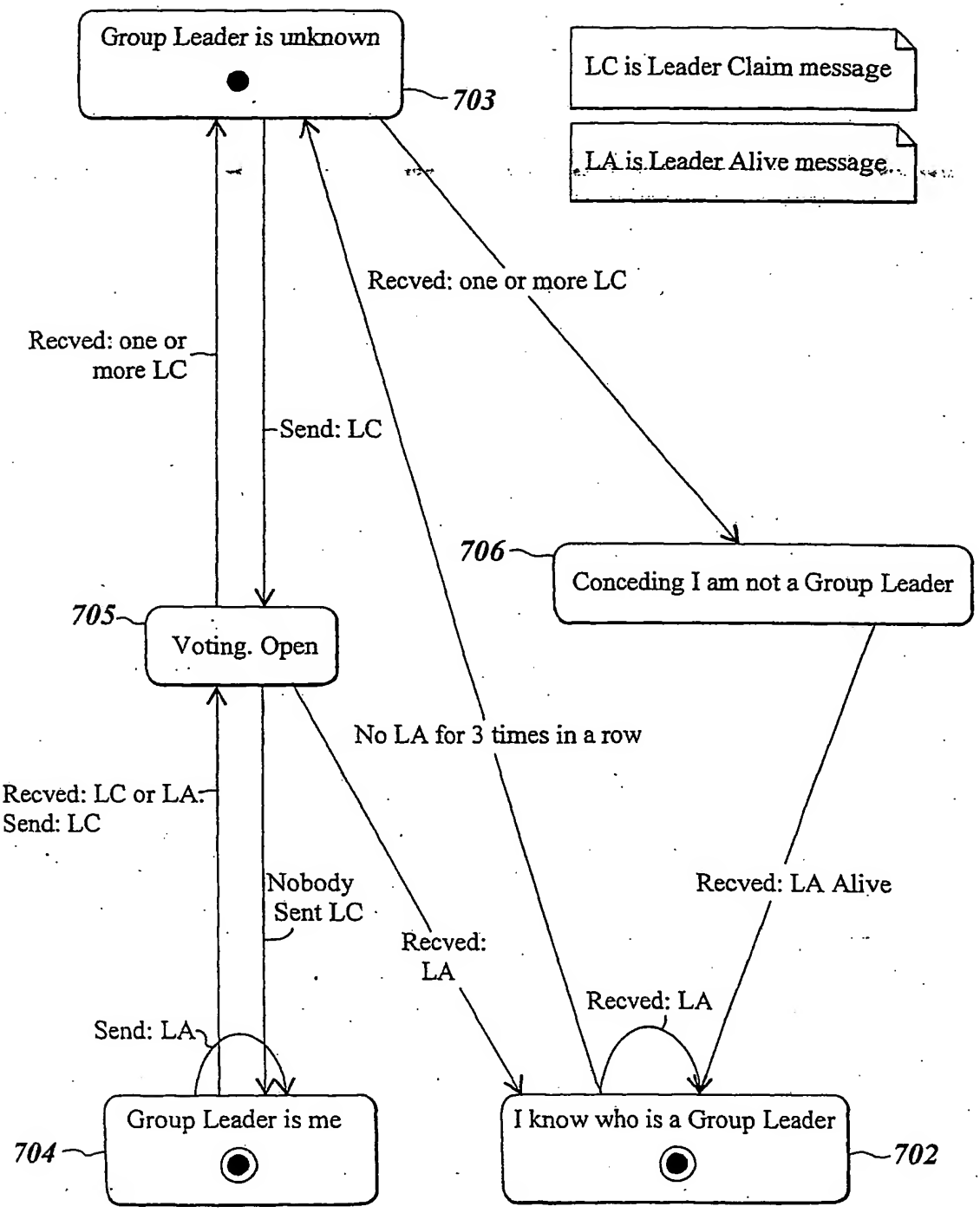


FIG. 5

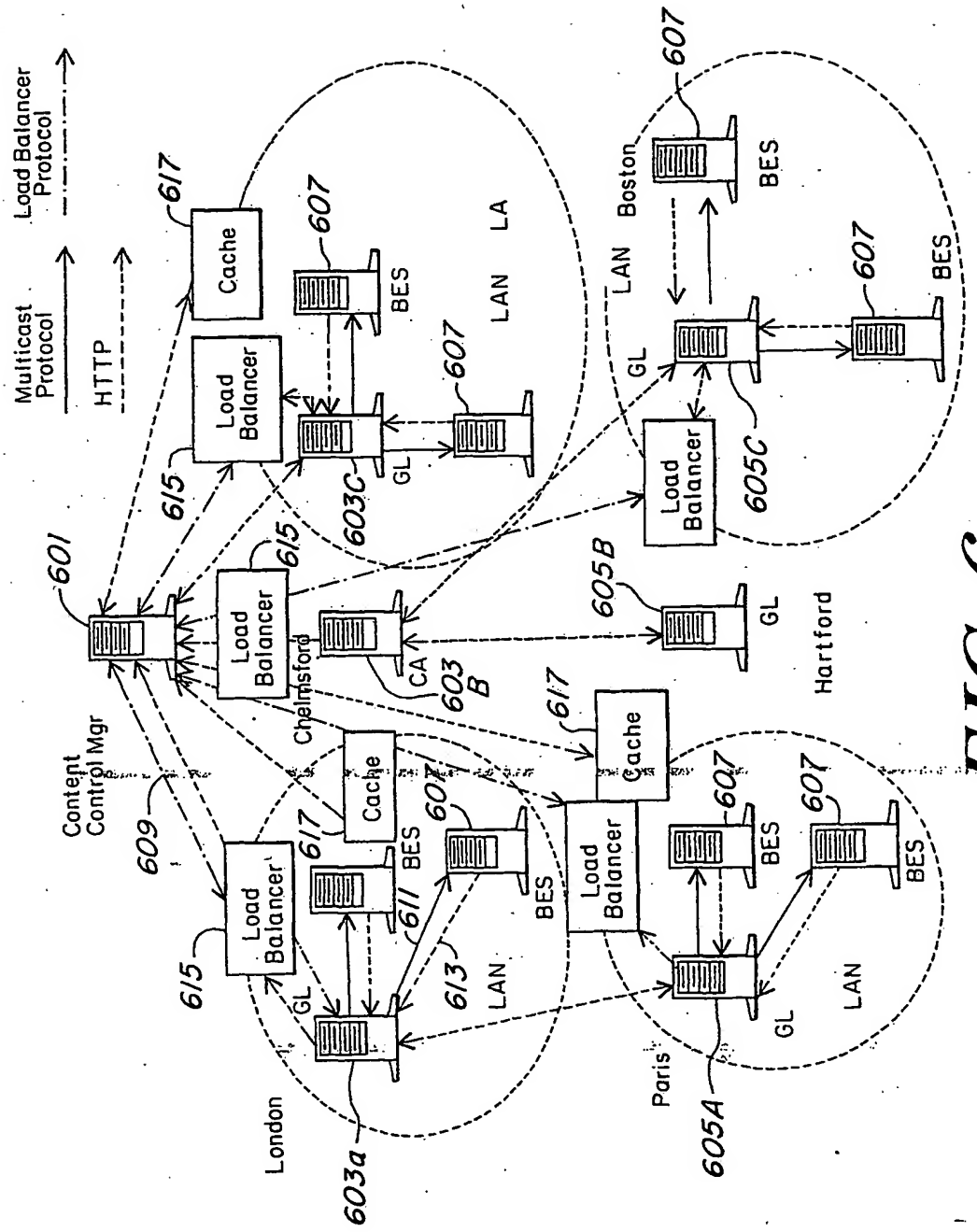


FIG. 6

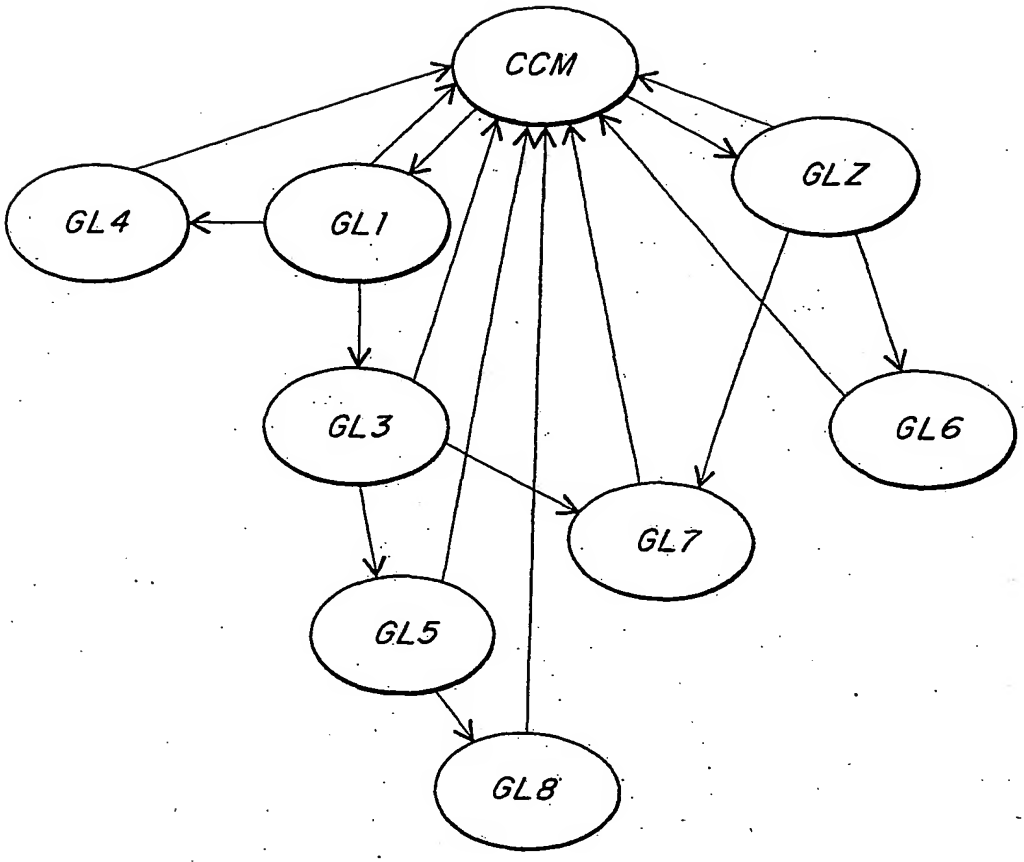


FIG. 7

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
18 October 2001 (18.10.2001)

PCT

(10) International Publication Number
WO 01/077841 A3

(51) International Patent Classification[7]: H04L 29/06

(21) International Application Number: PCT/US01/11505

(22) International Filing Date: 9 April 2001 (09.04.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

09/545,067	7 April 2000 (07.04.2000)	US
09/544,754	7 April 2000 (07.04.2000)	US
09/545,337	7 April 2000 (07.04.2000)	US

(71) Applicant: **NETWORK APPLIANCE, INC.** [US/US]; 495 East Java Drive, Sunnyvale, CA 94089 (US).

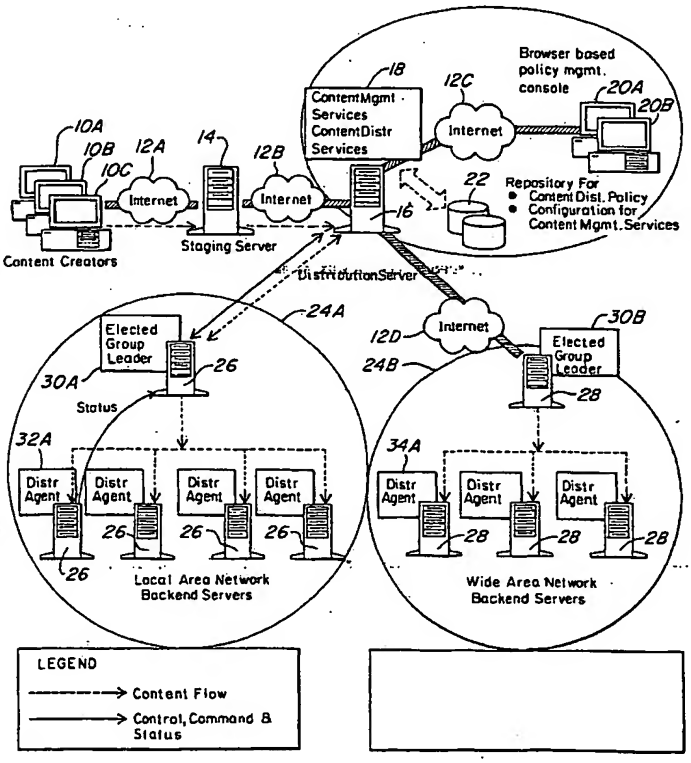
(72) Inventors: **BASANI, Vijay, R.**; 26 Kessler Farm Drive, #418, Nashua, NH 03062 (US). **MANGIAPUDI, Krishna**; 5 Decatur Drive, Nashua, NH 03062 (US). **MURACH, Lynne, M.**; Bramble Hill Road, Methuen, MA 01844 (US). **KARGE, Leroy, R.**; 400 White Pond Road, Leominster, MA 01453 (US). **REVSIN, Vitaly, S.**; 4 Enfield Drive, Andover, MA 01810 (US). **BESTAVROS, Azer**; 46 Rice Road, Wayland, MA 01778 (US). **CROVELLA, Mark, E.**; 14 Collier Road, Scituate, MA 02066 (US). **LAROSA, Domenic, J.**; 16 Meditation Lane, Atkinson, NH 03062 (US).

(74) Agents: **NELSON, Barry, C.** et al.; Brown Rudnick Freed & Gesmer, One Financial Center, Boston, MA 021111 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

[Continued on next page]

(54) Title: METHOD AND APPARATUS FOR RELIABLE AND SCALABLE DISTRIBUTION OF DATA FILES IN DISTRIBUTED NETWORKS



(57) Abstract: The present invention provides a system and apparatus for efficient and reliable, control and distribution of data files or portions of files, applications, or other data objects in large-scale distributed networks. A unique content-management front-end provides efficient controls for triggering distribution of digitized data content to selected groups of a large number of remote computer servers. Transport-layer protocols interact with distribution controllers to automatically determine an optimized tree-like distribution sequence to group leaders selected by network devices at each remote site. Reliable store-and-forward transfer to clusters is accomplished using a unicast protocol in the ordered tree sequence. Once command messages and content arrive at all participating group leaders, local hybrid multicast protocols efficiently and reliably distribute them to the back-end nodes for interpretation and execution. Positive acknowledgement is then sent back to the content manager from each group leader, and the updated content in each remote device autonomously goes "live" when the content change is locally completed.

WO 01/077841 A3

(84) **Designated States** *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*

(88) **Date of publication of the international search report:**
6 February 2003

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

XOCID: <WO___0177841A3_I_>

INTERNATIONAL SEARCH REPORT

International Application No
PCT/US 01/11505

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 H04L29/06

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 H04L G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)
EPO-Internal, WPI Data, PAJ, IBM-TDB, INSPEC, COMPENDEX

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X Y	WO 00 19291 A (ERICSSON TELEFON AB L M) 6 April 2000 (2000-04-06) abstract page 2, line 14 -page 4, line 20 page 6, line 3 -page 7, line 25 page 8, line 30 -page 10, line 3 page 11, line 7 - line 20	1,6 4,5, 13-16,19
Y	US 5 809 287 A (SHAFFER DAVID SCOTT ET AL) 15 September 1998 (1998-09-15) abstract column 3, line 13 -column 4, line 45 column 10, line 30 - line 34	5,14-16

-/--

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents; such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
4 September 2002	23. 09. 2002

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL – 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Peeters, D

Form PCT/ISA/210 (second sheet) (July 1992)

International Application No
PCT/US 01/11505

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 5 920 701 A (CATES KENNETH ET AL) 6 July 1999 (1999-07-06) cited in the application abstract column 4, line 35 -column 5, line 14	4,13
Y	US 5 699 501 A (CHANDRA TUSHAR DEEPAK ET AL) 16 December 1997 (1997-12-16) cited in the application abstract	19
X	US 5 390 326 A (SHAH NITIN J) 14 February 1995 (1995-02-14)	25,27, 30, 32-34,37
Y	abstract column 2, line 17 - line 37 column 3, line 55 -column 4, line 63 column 6, line 64 -column 7, line 5 column 16, line 42 -column 17, line 58	26,37
X	EP 0 598 674 A (IBM) 25 May 1994 (1994-05-25) abstract column 2, line 12 - line 25 column 5, line 36 -column 7, line 32 column 8, line 45 - line 46	35
Y	WO 99 55041 A (KADANSKY MIRIAM C ;ROSENZWEIG PHILIP M (US); CHIU DAH MING (US); W) 28 October 1999 (1999-10-28) abstract page 11, line 20 -page 13, line 12	26,37
A	WO 95 15635 A (YADEGAR JACOB ;PUROHIT BHARAT (GB); BRITISH TELECOMM (GB); BUSUIOC) 8 June 1995 (1995-06-08) abstract page 1, line 1 - line 13 page 18, line 10 -page 21, line 26	37
A	WO 98 48343 A (MOTOROLA INC) 29 October 1998 (1998-10-29) page 2, line 7 -page 5, line 17; figure 1	38-44

-/--

DOCID: <WO__0177841A3_1_>

International Application No
PCT/US 01/11505

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	LI GONG ET AL: "Elements of trusted multicasting" NETWORK PROTOCOLS, 1994. PROCEEDINGS., 1994 INTERNATIONAL CONFERENCE ON BOSTON, MA, USA 25-28 OCT. 1994, LOS ALAMITOS, CA, USA,IEEE COMPUT. SOC, 25 October 1994 (1994-10-25), pages 23-30, XP010100693 ISBN: 0-8186-6685-4 abstract page 24, paragraph 3.1 -page 25 page 26, line 15 - line 17	25-44
Y	OBRACZKA K ET AL: "A tool for massively replicating Internet archives: design, implementation, and experience" DISTRIBUTED COMPUTING SYSTEMS, 1996., PROCEEDINGS OF THE 16TH INTERNATIONAL CONFERENCE ON HONG KONG 27-30 MAY 1996, LOS ALAMITOS, CA, USA,IEEE COMPUT. SOC, US, 27 May 1996 (1996-05-27), pages 657-664, XP010167640 ISBN: 0-8186-7399-0	45,47
A	page 657, paragraph 1 -page 658 page 658, right-hand column, paragraph 2 -page 659, left-hand column page 663, line 33 - line 40	54
Y	FURHT B ET AL: "IP SIMULCAST: A NEW TECHNIQUE FOR MULTIMEDIA BROADCASTING OVER THE INTERNET" CIT. JOURNAL OF COMPUTING AND INFORMATION TECHNOLOGY, ZAGREB, HR, vol. 6, no. 3, September 1998 (1998-09), pages 245-254, XP000870379 ISSN: 1330-1136	45,47
A	abstract page 249, paragraph 3.2 -page 250	1
X	US 5 541 927 A (SABNANI KRISHAN K ET AL) 30 July 1996 (1996-07-30) abstract column 5, line 14 -column 6, line 9 column 9, line 25 - line 47	66

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US 01/11505

Box I Observations where certain claims were found unsearchable (Continuation of item 1 of first sheet)

This International Search Report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. [ ] Claims Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:

2. [ ] Claims Nos.:
because they relate to parts of the International Application that do not comply with the prescribed requirements to such an extent that no meaningful International Search can be carried out, specifically:

3. [ ] Claims Nos.:
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box II Observations where unity of invention is lacking (Continuation of item 2 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

see additional sheet

1. [X] As all required additional search fees were timely paid by the applicant, this International Search Report covers all searchable claims.

2. [ ] As all searchable claims could be searched without effort justifying an additional fee, this Authority did not invite payment of any additional fee.

3. [ ] As only some of the required additional search fees were timely paid by the applicant, this International Search Report covers only those claims for which fees were paid, specifically claims Nos.:

4. [ ] No required additional search fees were timely paid by the applicant. Consequently, this International Search Report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

Remark on Protest

[ ] The additional search fees were accompanied by the applicant's protest.

[X] No protest accompanied the payment of additional search fees.

FURTHER INFORMATION CONTINUED FROM PCT/ISA/ 210

This International Searching Authority found multiple (groups of) inventions in this international application, as follows:

1. Claims: 1-24

   A method and system for distributing information.

2. Claims: 25-44

   A method and system for determining a group leader, and a method for determining registration of its group members.

3. Claims: 45-68

   A method and system for distributing information to a set of destination nodes, and a method for determining completion status.

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No
PCT/US 01/11505

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 0019291 A	06-04-2000	US 6289511 B1	11-09-2001
		AU 1193500 A	17-04-2000
		EP 1145093 A2	17-10-2001
		WO 0019291 A2	06-04-2000
US 5809287 A	15-09-1998	US 5586304 A	17-12-1996
		US 5588143 A	24-12-1996
		US 5960189 A	28-09-1999
		AU 695638 B2	20-08-1998
		AU 3053895 A	21-03-1996
		CA 2157728 A1	09-03-1996
		EP 0703531 A1	27-03-1996
		JP 8227355 A	03-09-1996
US 5920701 A	06-07-1999	US 5727002 A	10-03-1998
		US 5553083 A	03-09-1996
		AU 4092597 A	19-03-1998
		WO 9809412 A1	05-03-1998
		US 6151696 A	21-11-2000
		AT 209411 T	15-12-2001
		AU 5295096 A	07-08-1996
		DE 69617204 D1	03-01-2002
		DE 69617204 T2	04-07-2002
		DK 804838 T3	18-03-2002
		EP 1128591 A2	29-08-2001
		EP 0804838 A2	05-11-1997
		ES 2163011 T3	16-01-2002
		JP 10512726 T	02-12-1998
		WO 9622641 A2	25-07-1996
US 5699501 A	16-12-1997	EP 0805393 A2	05-11-1997
		JP 10040227 A	13-02-1998
US 5390326 A	14-02-1995	NONE	
EP 0598674 A	25-05-1994	US 5365523 A	15-11-1994
		CA 2100542 A1	17-05-1994
		EP 0598674 A1	25-05-1994
		JP 2502922 B2	29-05-1996
		JP 6350652 A	22-12-1994
WO 9955041 A	28-10-1999	US 6185698 B1	06-02-2001
		AU 3196899 A	08-11-1999
		EP 1074115 A1	07-02-2001
		JP 2002512483 T	23-04-2002
		WO 9955041 A1	28-10-1999
WO 9515635 A	08-06-1995	AU 692810 B2	18-06-1998
		AU 1113295 A	19-06-1995
		AU 701581 B2	04-02-1999
		AU 6476198 A	02-07-1998
		CA 2177488 A1	08-06-1995
		CA 2318582 A1	08-06-1995
		CN 1136873 A	27-11-1996
		EP 0732018 A1	18-09-1996
		WO 9515635 A1	08-06-1995
		JP 2935987 B2	16-08-1999
		JP 10294770 A	04-11-1998

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No
PCT/US 01/11505

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 9515635 A		JP 2971580 B2	08-11-1999
		JP 9505917 T	10-06-1997
		NZ 276747 A	26-06-1998
		NZ 330166 A	28-01-2000
		SG 47804 A1	17-04-1998
		US 6226273 B1	01-05-2001
		US 2001033551 A1	25-10-2001
WO 9848343 A	29-10-1998	AU 735576 B2	12-07-2001
		AU 6544098 A	13-11-1998
		BR 9815478 A	06-11-2001
		CN 1253641 T	17-05-2000
		EP 1029263 A1	23-08-2000
		JP 2001521716 T	06-11-2001
		WO 9848343 A1	29-10-1998
		US 2001014652 A1	16-08-2001
US 5541927 A	30-07-1996	CA 2151072 A1	25-02-1996
		EP 0698975 A2	28-02-1996
		JP 8088633 A	02-04-1996